



SANDIA REPORT

SAND2003-8782

Unlimited Release

Printed January 2004

Numerical Tools for Atomistic Simulations

P.M. Gullett, G.Wagner, A. Slepoy

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.doe.gov/bridge>

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: <http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online>



SAND2003-8782
Unlimited Release
Printed January 2004

Numerical Tools for Atomistic Simulations

P.M. Gullett, G.J. Wagner
Center for Materials and Engineering Sciences
Sandia National Laboratories
Livermore, CA 94551

A. Slepoy
Computation, Computers & Math Center
Sandia National Laboratories
Albuquerque, NM 87123

M.F. Horstemeyer, H. Fang
Department of Mechanical Engineering
Mississippi State University
Mississippi State, MS 39762

Mo Li
Materials Science and Engineering
771 Ferst Drive, N.W.
Atlanta, GA 30332-0245

M.I. Baskes
Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM 87545

Abstract

The final report for a Laboratory Directed Research and Development project entitled “Parallel Atomistic Computing for Failure Analysis of Micromachines” is presented. In this project, atomistic algorithms for parallel computers were developed to assist in quantification of microstructure-property relations related to weapon micro-components. With these and other serial computing tools, we are performing atomistic simulations of various sizes, geometries, materials, and boundary conditions. These tools provide the capability to handle the different size-scale effects required to predict failure. Nonlocal continuum models have been proposed to address this problem; however, they are phenomenological in nature and are difficult to validate for micro-scale components. Our goal is to separately quantify damage nucleation, growth, and coalescence mechanisms to provide a basis for macro-scale continuum models that will be used for micro-machine design. Because micro-component experiments are difficult, a systematic computational study that employs Monte Carlo methods, molecular statics, and molecular dynamics (EAM and MEAM) simulations to compute continuum quantities will provide mechanism-property relations associated with the following parameters: specimen size, number of grains, crystal orientation, strain rate, temperature, defect nearest neighbor distance, void/crack size, chemical state, and stress state. This study will quantify size-scale effects from nanometers to microns in terms of damage progression and thus potentially allow for optimized micro-machine designs that are more reliable and have higher fidelity in terms of strength. In order to accomplish this task, several atomistic methods needed to be developed and evaluated to cover the range of defects, strain rates, temperatures, and sizes that a material may see in micro-machines. Therefore we are providing a complete set of tools for large scale atomistic simulations that include pre-processing of realistic material configurations, processing under different environments, and post-processing with appropriate continuum quantities. By running simulations with these tools, we are able to determine size scale effects that correlate microstructure and defect configurations with mechanical properties of materials.

Keywords: Parallel computing, poly-crystalline, dislocations, atomistic simulation.

Contents

1	Introduction	9
1.1	Prewarp	10
1.2	EAMPM	10
1.3	Updated WARP	10
2	Prewarp	11
2.1	Concepts	11
2.1.1	System	11
2.1.2	Grains	12
2.1.3	Lattices	14
2.1.4	Primitive Vectors	14
2.1.5	Atom Positions	14
3	Molecular Statics	18
3.1	Introduction	18
3.2	Embedded Atom Method	18
3.2.1	Motivation	18
3.2.2	Formalism	19
3.2.3	Difficulties with non-pairwise interactions	20
3.3	Parallel strategy	20
3.3.1	Motivation	20
3.3.2	Spatial decomposition	20
3.3.3	Checkerboard updating and reduced buffer algorithms	22
3.3.4	Reduced communication scheme	22
3.4	Embedded Atom Method particulars	23
3.4.1	Two-way communication	23
3.4.2	Domain size restriction	24
3.5	Scaling study	24
3.5.1	Fixed system size	24
3.5.2	Equal load scaling	24

4	Molecular Dynamics: the Modified Embedded Atom Method Potential in Warp	27
4.1	The MEAM Potential	27
4.1.1	Embedding energy potential	28
4.1.2	Pair potential	29
4.1.3	Screening function	29
4.2	Notes on Notation	30
4.3	Derivatives of the Screening Function	31
4.4	Derivatives of the Electron Densities	32
4.4.1	Zeroth-order density	33
4.4.2	Higher-order densities	33
4.4.3	Derivatives of the averaged weighting factors	35
4.4.4	Derivatives of Γ_i	36
4.4.5	Derivative of the total density	36
4.5	Derivatives of the MEAM Energy	37
5	Postprocessing	40
5.1	Introduction	40
5.2	Strain Formulation	41
5.2.1	Continuum Deformation Gradient	41
5.2.2	Strain Tensors	42
5.2.3	Discrete Deformation Gradient	42
5.2.4	Weight Function	45
5.2.5	Discussion	45
5.2.6	Example Strain Calculation	46
6	Computational Example	52
6.1	Introduction	52
6.2	Problem Statement	52
6.2.1	Atomic lattice	52
6.2.2	Boundary conditions	52
6.2.3	Initial conditions	53
6.2.4	Run cases	53
6.2.5	Data collection	54
6.3	Results	55
6.4	Discussion	56
7	Bibliography	64
8	Distribution	67

List of Figures

2.1	Real and simulated microstructures	12
2.2	Polycrystalline lattices at various size scales	13
2.3	fcc and bcc lattice structures generated with Prewarp	15
2.4	Quartz structures generated with Prewarp	16
2.5	HCP lattice structures generated with Prewarp. In these figures, the (0001) planes are parallel to the xy-plane.	17
3.1	Embedding densities of neighbor atoms.	19
3.2	2d representation of a domain decomposition topology.	21
3.3	Simulation times for a fixed system size and variable number of processors.	25
3.4	Equal load scaling	26
5.1	Local motion of an atomic body	43
5.2	Example weight function for strain computation	46
5.3	Schematic of shear simulation	47
5.4	The computed average local Green and Almansi shear strains along with the average shear stress are plotted vs applied strain.	49
5.5	The percent difference in the computed average local Green and Almansi shear strains are plotted vs applied strain.	50
5.6	Almansi strain at 120 picoseconds	51
5.7	Green strain at 120 picoseconds	51
6.1	Geometry used for uniaxial tension void growth tests.	53
6.2	Development of dislocation structure for run 1.	57
6.3	Development of dislocation structure for run 2.	58
6.4	Development of dislocation structure for run 3.	59
6.5	Development of dislocation structure for run 4.	60
6.6	Stress-strain behavior of the four simulations.	61
6.7	Void fraction vs. strain.	62
6.8	Void fraction vs. plastic strain.	63

List of Tables

6.1	Summary of cases run	54
-----	--------------------------------	----

Chapter 1

Introduction

In this project, atomistic algorithms for parallel computers were developed to assist in quantification of microstructure-property relations related to weapon micro-components. With these and other serial computing tools, we are performing atomistic simulations of various sizes, geometries, materials, and boundary conditions. These tools provide a capability to handle the different size-scale effects required to predict failure. Nonlocal continuum models have been proposed to address this problem; however, they are phenomenological in nature and are difficult to validate for micro-scale components. Our goal is to separately quantify damage nucleation, growth, and coalescence mechanisms to provide a basis for macro-scale continuum models that will be used for micro-machine design. Because micro-component experiments are difficult, a systematic computational study that employs Monte Carlo methods, molecular statics, and molecular dynamics (Embedded Atom Method and Modified Embedded Atom Method) simulations to compute continuum quantities will provide mechanism-property relations associated with the following parameters: specimen size, number of grains, crystal orientation, strain rate, temperature, defect nearest neighbor distance, void/crack size, chemical state, and stress state. This study will quantify size-scale effects from nanometers to microns in terms of damage progression and thus potentially allow for optimized micro-machine designs that are more reliable and have higher fidelity in terms of strength. In order to accomplish this task, several atomistic methods required development and evaluation to cover the range of defects, strain rates, temperatures, and sizes that a material may see in micro-machines. Therefore we are providing a complete set of tools for large scale atomistic simulations that include pre-processing of realistic material configurations, processing under different environments, and post-processing with appropriate continuum quantities. By running simulations with these tools, we are able to determine size scale effects that correlate microstructure and defect configurations with mechanical properties of materials.

This report presents some details regarding the computer codes developed or enhanced for this project including Prewarp, EAMPM, and Warp.

1.1 Prewarp

The pre-processing code (Prewarp) is an atomic lattice configuration generating program. The input for prewarp is easy to use because it is organized around a few simple concepts. It was written in C++ and should be portable to most computing platforms.

Prewarp was developed to provide software capable of generating atomic system configurations with a variety of defects, grain orientations, and grain boundaries. Prewarp includes several simple crystal lattice types, but allows for user input of complex multi-species lattice structures through specification of lattice primitive vectors and basis vectors.

1.2 EAMPM

EAMPM (Embedded Atom Method Parallel Monte Carlo) is a parallel Monte Carlo code. We present the first scalable massively parallel version of a Monte Carlo simulation using the Embedded Atom Method (EAM) potential. This code is a valuable tool for studies of atomistic systems and the bulk properties of metals that are sensitive to the finite size effects at the size scale of the smaller serial simulation. While EAM parallel dynamics codes exist, the explicit treatment of time and full trajectory information limit them to the real time scales of nanoseconds. The Monte Carlo technique permits us to study the equilibrium properties of the system. The technique also provides a platform for overcoming the energetic barriers, which can effectively trap a dynamics simulation. We expect the simulation to be particularly effective at treating multi-component systems, such as alloys, by allowing an identity swap as a Monte Carlo update.

1.3 Updated WARP

Warp is a parallel molecular dynamics simulation code for modeling stress and strain in materials that originally implemented the embedded atom method (EAM) and Lennard-Jones (LJ) potentials. It is written in F90 and performs message-passing via MPI calls. It has been used for simulations involving up to 100 million atoms [1], and has the capability of easily applying deformation paths to such as tension, shear, and torsion boundary conditions [2]. Warp has been modified in this project to include the modified embedded atom method (MEAM) potential, multi-species simulation capability and constant pressure equilibration.

Chapter 2

Prewarp

The pre-processing code (Prewarp) is an atomic lattice configuration generating program. The input for prewarp is easy to use because it is organized around a few simple concepts. It was written in C++ and should be portable to most computing platforms.

Prewarp was developed to provide software capable of generating atomic system configurations with a variety of defects, grain orientations, and grain boundaries. Prewarp includes several simple crystal lattice types, but allows for user input of complex multi-species lattice structures through specification of lattice primitive vectors and basis vectors.

The key concepts used in prewarp to define an atomic configuration include the system, grain, and lattice objects, as well as configuration modifiers. These are similar to the domain, element and material type definitions in a finite element model. The system is a region that will contain the complete atomic configuration; it includes a boundary and contains a user-specified set of grain definitions. Grains are objects that provide a spatial decomposition of the system, much like finite elements. They are defined by a location, lattice definition, and lattice orientation. A grain may or may not include explicit specification of a boundary. A lattice definition is a general framework that contains the information necessary to “build” a crystal lattice structure.

2.1 Concepts

2.1.1 System

The system is a container that provides the framework for assembling a set of atomic positions that is consistent with a polycrystalline lattice structure. It is constructed by supplying its geometric extents, including periodic boundaries, and then defining grains, lattice types and defects. Once these definitions are in place the atomic configuration can be built. Building the lattice proceeds by first building lattice types defined for the system, and then adding lattices to the system based on the geometry and orientation of the grains. After construction of the system, other changes can be made such as adding or replacing atoms and/or introducing defect structures.

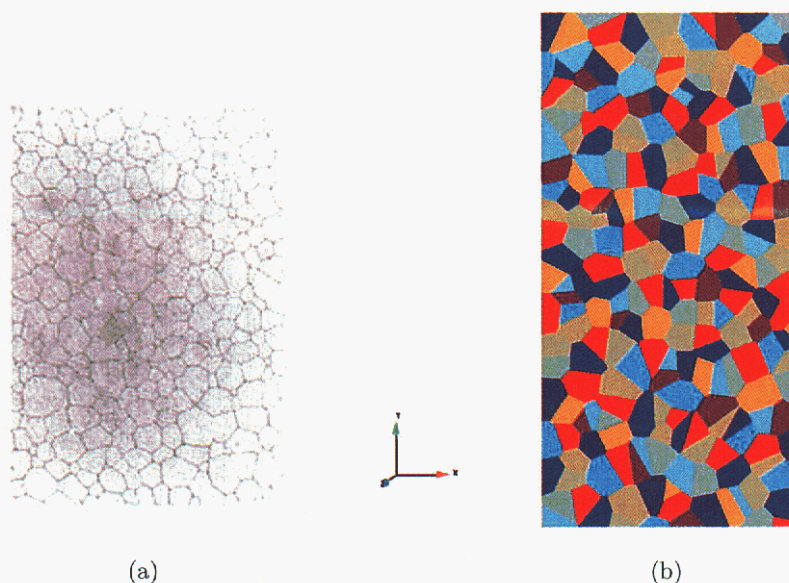


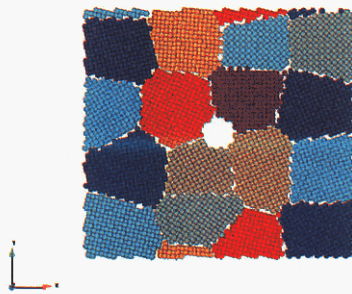
Figure 2.1: On the left is an example of a real microstructure of polycrystalline Ni. On the right is an example of a Prewarp polycrystalline structure with 1.6 million atoms. The colors indicate individual grains.

2.1.2 Grains

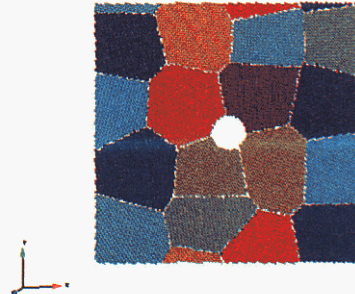
In Prewarp, a grain provides a means of generating a spatial decomposition of the system into regions with different crystal lattice structures. A grain is defined by specification of its position, lattice type, lattice orientation, and (optionally) a bounding surface.

The grain position, or site, is the location from which the region occupied by the grain is to be determined. How the region is determined depends on whether a boundary is specified. If a boundary is specified, such as a sphere, then the spherical region is centered at the site. If no boundary is specified, then the region occupied by the grain will be determined by the relative positions of other grain sites. This latter procedure generates a voronoi decomposition of the system, and is useful for generating configurations that are similar to polycrystalline materials as shown in figure 2.1. This lattice is composed of 255 grains and includes 1.6 million atoms.

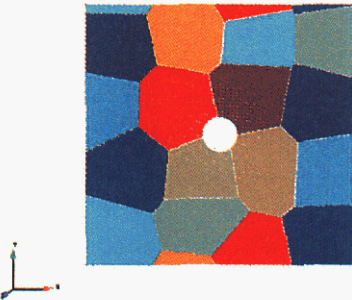
For easy specification of numerous grains, Prewarp provides automatic grain generation of grains. Grain sites can be generated using a grid-based or random site location approaches. The grid-based system divides the system block into a number of subregions, placing grain sites within the the center of the blocks. This procedure is similar to that described in [3]. Orientation of grains may be specified as a random normal distribution of angles in two or three dimensions about a primary axis. The distribution angles can be limited by specifying a maximum angle.



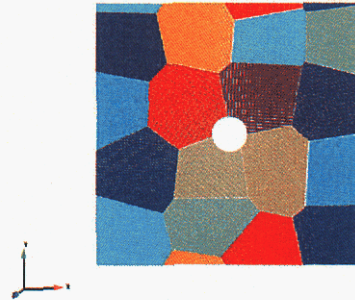
(a) 11,000 atoms



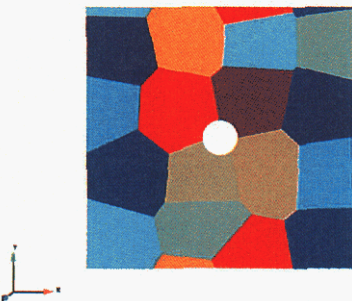
(b) 42,000 atoms



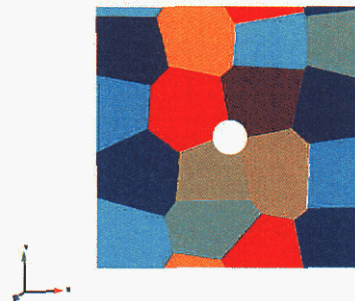
(c) 172,000 atoms



(d) 676,000 atoms



(e) 3,610,000 atoms



(f) 14,475,000 atoms

Figure 2.2: Polycrystalline lattices with a single void are shown for various size scales. The systems are periodic in the $y - dir$ and individual grains are distinguished by color. The atomic positions correspond to perfect lattice positions.

2.1.3 Lattices

Prewarp utilizes a lattice type class that stores the information needed to create most atomic crystalline lattice structures. Several built-in lattice structures are available; however, user-specified lattices can be included as well. A user lattice is specified by defining its unit cell which includes translation vectors for the lattice points and atom coordinates in terms of basis vectors. The grain crystal structure is created by the translation of the unit cell along the primitive vectors.

2.1.4 Primitive Vectors

Unit cells are specified by three vectors primitive vectors that lie along the cell edges. This specification includes lattice points only at the corners of the unit cell. The lengths of the vectors are the unit cell dimensions, and their directions are the crystallographic axes.

2.1.5 Atom Positions

The contents of the unit cell are defined by specification of the number, type and positions of atoms in the lattice relative to the lattice points. In Prewarp, a unit cell may contain any number of atoms and any number of atom types. Thus, Prewarp is able to construct crystalline alloy structures. The atomic positions in the unit cell are specified in terms of basis vectors, which are unit vectors that lie along the lattice primitive directions. The position of each atom is input by three coordinate values in terms of the basis vectors.

The simplest arrangements of atoms, applicable to many types of lattice structures *e.g.* single species scp, bcc, and fcc, is a single single atom located at the vertex of the unit cell $(0, 0, 0)$. Specification of unit cells in terms of primitive vectors and atom coordinates allows for generation of many complex lattice types with Prewarp. Several examples are shown in figures 2.3 - 2.5. Definitions of many types of structures are available in many common references such as [4] and on the web [5].

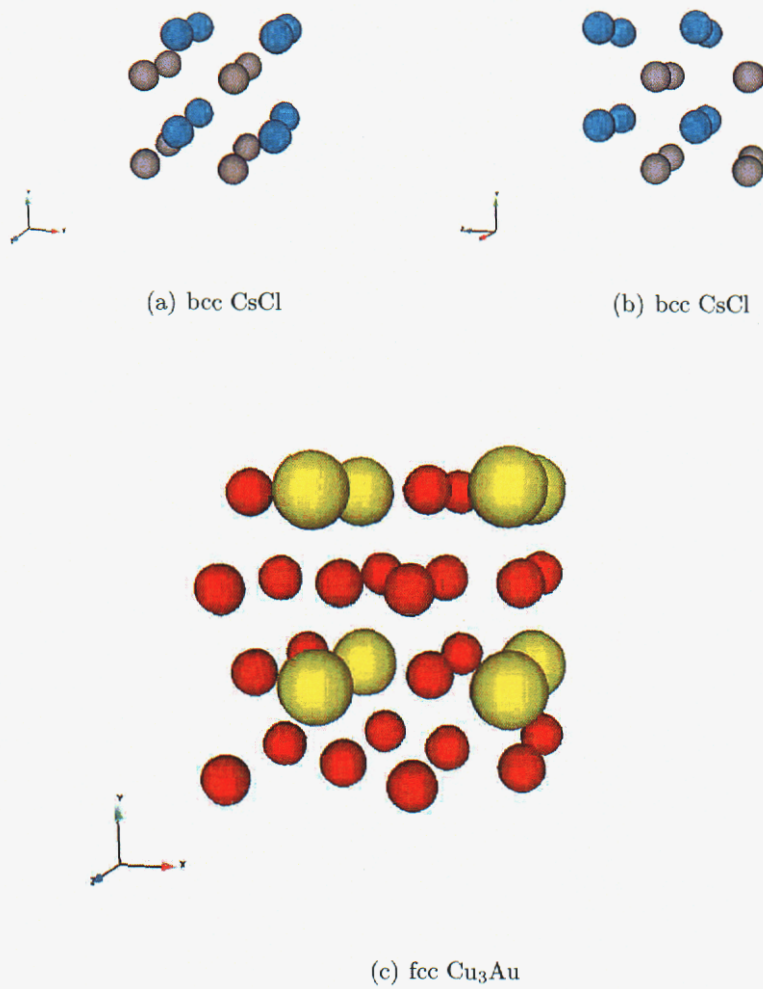
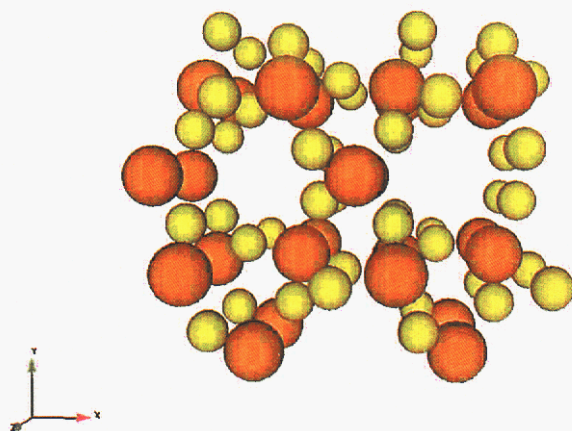
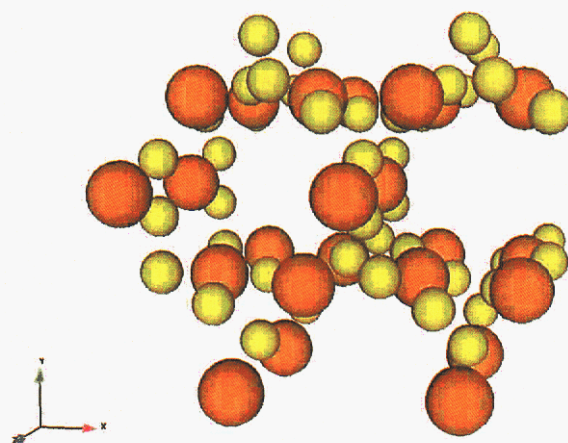


Figure 2.3: fcc and bcc lattice structures generated with Prewarp

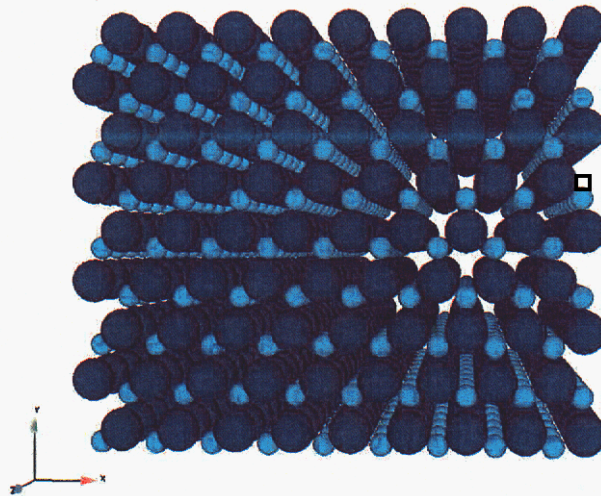


(a) α -Quartz



(b) β -Quartz

Figure 2.4: Quartz structures generated with Prewarp



(a) Tungsten carbide, WC

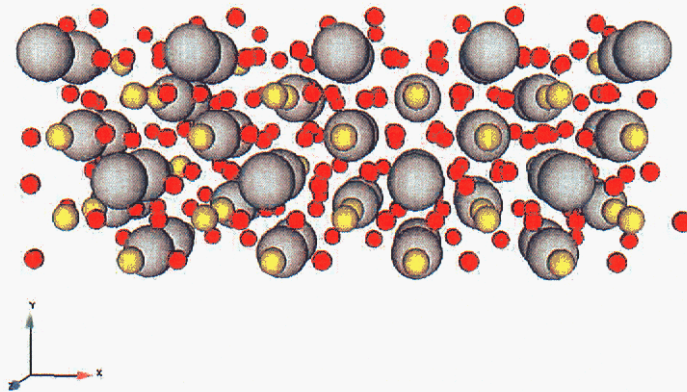
(b) Ilmenite, FeTiO_3

Figure 2.5: HCP lattice structures generated with Prewarp. In these figures, the (0001) planes are parallel to the xy-plane.

Chapter 3

Molecular Statics

3.1 Introduction

We present the first scalable massively parallel version of a Monte Carlo simulation using the Embedded Atom Method potential. This code is a valuable tool for studies of atomistic systems and the bulk properties of metals that are sensitive to the finite size effects at the size scale of the smaller serial simulation.

While EAM parallel dynamics codes exist, the explicit treatment of time and full trajectory information limit them to the real time scales of nanoseconds. The Monte Carlo technique permits us to study the equilibrium properties of the system. The technique also provides a platform for overcoming the energetic barriers, which can effectively trap a dynamics simulation.

We expect the simulation to be particularly effective at treating multi-component systems, such as alloys, by allowing an identity swap as a Monte Carlo update. Overall, we expect this simulation to serve as an effective bridge between the atomistic and meso-scales in describing the properties of pure metals and metal alloys.

We have verified the code by comparisons with the serial Monte Carlo version and by performing a scaling study. We intend to run a set of simulations to obtain a static stress-strain relationship for a large atomistic system, accessible only with the massively parallel simulation. The simulation box size is critical to limit the finite size effects and to reduce fluctuations in the stress averages.

3.2 Embedded Atom Method

3.2.1 Motivation

For all elements with unfilled valence shells, an approximation of non-interacting bond strengths fails with various degrees of magnitude. The problem with assuming that the bond strength is independent of the coordination number lies with the fact that removing neighbors strengthens the remaining bonds [6]. So any pairwise parametrization that

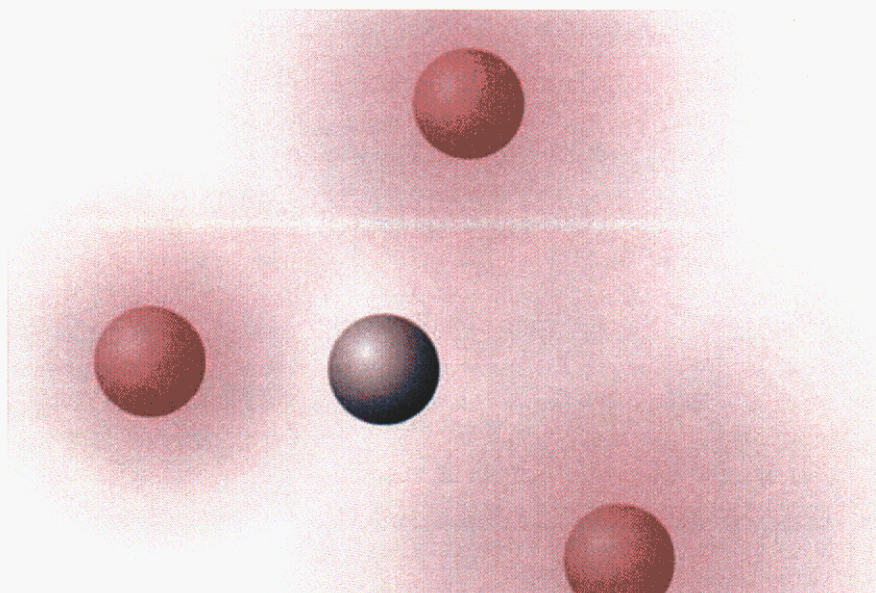


Figure 3.1: The energy of an atom (center) is given by the sum of its pairwise interactions with its neighbors and by a functional of the combined neighbor densities at the atom's location.

works for the full coordination in the bulk will break down near the surface. To deal with this problem, a many-body parametrization model is needed. The EAM [7] is among the most successful such models, that also enjoys the advantage of retaining the locality of the potential and the computational efficiency of a pairwise potential.

3.2.2 Formalism

The EAM [7] uses a cumulant of effective embedding densities to correct for the deficiencies of its pairwise terms. Then the energy of a given atom is given by the sum of its pairwise interactions with neighbors and a correction term that is a function of the combined effective neighbor densities.

$$E = \sum_{i,j} \Phi(R_{i,j}) + \sum_i F \left(\sum_j \rho(R_{i,j}) \right) \quad (3.1)$$

This later contribution has a many-body flavor, while it still remains local. Both the pairwise interaction strength and the density dependent correction are parametrized to fit experimental properties. Figure 3.1 illustrates the computational format of the potential. The energy of the atom in the center is given by the sum of its pair-wise interactions with the neighboring atoms and by the embedding energy function evaluated at the cumulative density value contributed by the same neighboring atoms.

3.2.3 Difficulties with non-pairwise interactions

The many-body character of the interaction introduces an added level of complexity. In a “pair-wise only” interaction, all the information needed to evaluate energy *changes* due to an update of a single atom position is contained in the positions of the neighboring atoms within a certain cutoff radius. The extra correction term requires information of the effective embedding densities at the positions of these atoms as well, since no information is given about their respective neighbors that lie outside the cutoff radius.

In a serial version of the code, the information of all positions and all densities is always accessible. Therefore, in a serial version of the an energy evaluation, the need for extra information can easily be overcome by extending the radius of the neighbor list membership to twice the cutoff radius, or by simply referring to a stored list of the respective cumulative densities. In a parallel implementation, however, where information is not always local and information passing is often the bottleneck, this issue can become critical.

3.3 Parallel strategy

3.3.1 Motivation

Computation speed and memory limitations constrain us to small system sizes which often exhibit strong finite size effects. The push for larger system size has been partially addressed by parallel computing strategies. For some computational problems, parallel processing has paid off with high efficiency returns. Among these successes, bulk atomistic simulations with local potentials approach closely perfect scaling efficiency of N/P where N is the total number of atoms in the system, and P is the total number of processors. Excellent spatial domain decomposition (DD) algorithms [8] make this possible. And, while several other methods have been introduced to deal with intermediate size non-homogeneous systems [9], for very large bulk systems DD remains the algorithm of choice.

3.3.2 Spatial decomposition

The power of the spatial decomposition algorithm is in limiting all run time communications to purely local ones [8]. While all-to-all communication schemes become exceedingly expensive for more than a hundred or so processors, local communication simulations scale indefinitely at run-time, making it possible to simulate systems of tens of millions of atoms. This property of local potentials is the key to a truly parallel process where operations on the remote sections of the simulation can be carried out in complete independence of each other.

The DD algorithm partitions the simulation box into spatial domains of the 3 dimensional parallelepiped shape. It then assigns all the atoms within a given domain to an appropriate processor. This processor then updates its portion of the system using data structures that contain information about its own atoms and the relevant information about the atoms in

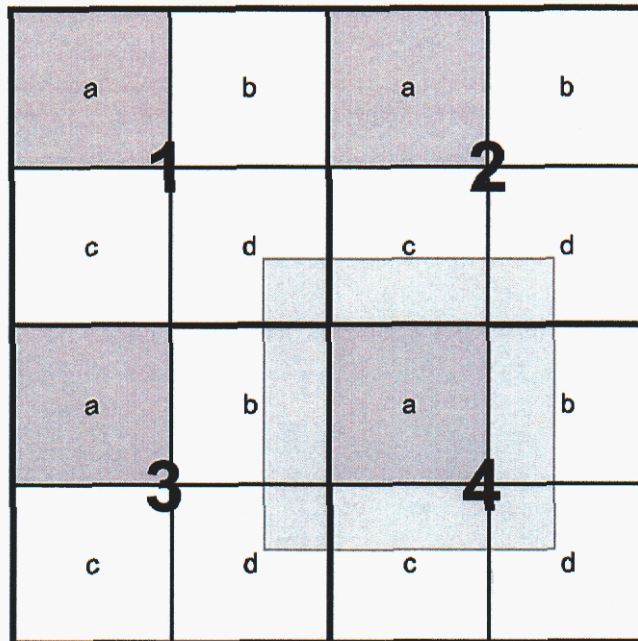


Figure 3.2: Domains are represented by numerals, while sub-domains are labeled by lower case letters. The dark shaded sub-domains are allowed simultaneous parallel update regions, while the light shading represents region that supplies information for the update of the enclosed sub-domain.

the spatially proximate domains. All the necessary information to make update decisions is obtained from neighboring processors alone. Figure 3.2 is a two-dimensional representation of a decomposition topology.

3.3.3 Checkerboard updating and reduced buffer algorithms

In the spirit of true parallelism, where operations on different portions of the simulation can be carried out independently, one must be certain that the updates on individual processors do not interact with each other. This is not guaranteed by topology, since neighboring processors can potentially operate on spatially adjacent regions of the simulation. To prevent this, a checkerboard updating scheme was developed [10]. In this scheme, processors operate on identical sub-domains which are proximately removed from their counterpart sub-domains on the neighboring processors. An example of such a topology is a square domain subdivided into quadrants. Figure 3.2 contains an example of such a decomposition. A parallel update is performed in the same quadrant of each processor's domain. This method introduces a buffer of space around each active region of the simulation, guaranteeing lack of interaction during the update. Once the update is complete, a subsequent communication of the changes allows updates to be carried out in the next set of sub domains.

The checkerboard scheme of updating produces another unexpected benefit. Since the update is carried out in a particular region only, the neighbor-neighbor communication can be reduced from all nearest neighbors to a small subset of the relevant neighbors only. In Figure 3.2, this effect is represented by the light shading of the region adjacent to Processor 4. It becomes apparent that the only processors that need to communicate information to Processor 4 are numbers 1, 2, and 3. If the grid extended further to include more domains adjacent to 4, those processors would not need to communicate. In two dimensions, the number of relevant neighboring processors is reduced from 8 to 3. In a 3-dimensional simulation, the reduction is from 26 to 7.

This is a particularly efficient process for an atomistic simulation with local interactions since only the coordinates of atoms within the potential cutoff layer need to be passed to the neighboring processors. The layer contains a small portion of all the atoms belonging to a given processor. This is represented in the Figure 3.2 by a light-shaded band around Processor 4. This produces several levels of speed-up. Smaller communication buffers lead to less communication time. Also, smaller neighbor arrays lead to smaller loops at processing times.

3.3.4 Reduced communication scheme

Communication times are often the bottleneck of a parallel simulation. To balance the computational time with the respective amount of communication, careful attention must be given to the communication process. Each communication period is composed of a latency period, and the actual transfer time. Unless the communicated buffer is very large, the communication time is dominated by the latency period. With a bit of a computational

expense, in three dimensions with the checkerboard updating, the information can be passed to the 7 relevant processors in only three explicit communications, thus reducing the number of latency periods to 3 instead of 7. This savings is accomplished by indirect passing of the communication buffers. For illustration, picture a two-dimensional system where updates occur in respective quadrants. An update in the upper left quadrant would need to be communicated to the processor above (north), the processor to the left (west), and the processor in the corner (north-west). Instead of passing buffers to each of these, we have all processors pass north. Then the receiving processors store the buffer received in addition to their own and subsequently pass west. This causes the processor in the north-west to receive the buffer it was sent from the south-east indirectly in the second communication. In two dimensions, the reduction is only from 3 to 2 communications. In higher dimensions, the improvement is significantly larger.

3.4 Embedded Atom Method particulars

3.4.1 Two-way communication

The EAM potential, though computationally local, is not pair-wise. The many-body aspects of this potential create complications when they interact with the spatial decomposition checkerboard update algorithm. For a pair-wise potential simulation, all the necessary external information to carry out an update in a given sub-domain is contained within the atomic positions of the relevant neighbor processors. For the EAM, an energy evaluation also requires the cumulative density of the same atoms. Since this quantity may depend on the atoms that are not in the cutoff range, additional communications are required.

To illustrate this point, picture two processors responsible for two adjacent spatial domains. In Figure 3.2, domains 2 and 4 share a north active boundary when an update is carried out in the sub-domain 4. To evaluate an energy change associated with this update, processor 4 needs atomic positions and cumulative densities of the subset of atoms on processor 2. These then are communicated prior to the update. The update in atomic positions on domain 4 sub-domain *a* leads to potential changes in the respective cumulative densities in the sub-domains *c* and *d* of domain 2. To update these densities, so that they can be reported correctly by domain 2 to the domain 4 for its subsequent update in sub-domain *b*, processor 2 needs to be informed of the nature of the update in *4a*. This then needs to be communicated by 4 to 2 at the end of the 1st update.

This logic is fairly complicated, especially when coupled with the checkerboard updates and the reduced communication scheme. In a nutshell, the algorithm uses two communications per update: before update, positions and densities of the relevant neighbor domains; after update, changes in positions to other relevant neighbor domains. Relevant neighbors for each sub-domain are predetermined and their identities stored in the initial setup.

3.4.2 Domain size restriction

The many-body nature of the potential also leads to an increased minimum size restriction on the domain size. To ensure region independence, we must be certain that energy changes due to individual parallel updates can be computed independently. This means that individual updates cannot cause two updated atomic positions from two separate domains to affect the same cumulative densities. This requires that the resultant spheres of the cutoff radius centered on their final positions cannot intersect. This leads to a minimum restriction of $2 * (c + m)$, where c is the potential cutoff and m is the maximum move length, on each dimension of the sub-domain, and twice that on each dimension of the entire domain.

3.5 Scaling study

The scaling study was conducted by running a simulation of a single atom type system through one thousand sweeps, where each sweep represents a single atom update attempt per sub-domain, or equivalently, eight atoms per processor. The simulation used periodic boundary conditions. The atoms were randomly placed. We used silver EAM potential for energy evaluations.

3.5.1 Fixed system size

A simulation of a fixed size of 35000 atoms was benchmarked running on various processor grid sizes. As shown in Figure 3.3, we have achieved nearly perfect scaling over the range of interest, where the runtime scales as $1/N$ with N being the number of processors. This scaling rate is indicative of the true parallelism where multiple tasks are performed simultaneously with no dependence on the total system size.

3.5.2 Equal load scaling

Another traditional test of parallel scaling properties is a simulation for a variable number of processors each with a constant load. Here we use 300 atoms per processor for 1000 sweeps. For perfect parallelism linear scaling, we can simply turn off the data analysis functions that require some all-to-all communication. The data, as shown, includes frequent sampling for analysis.

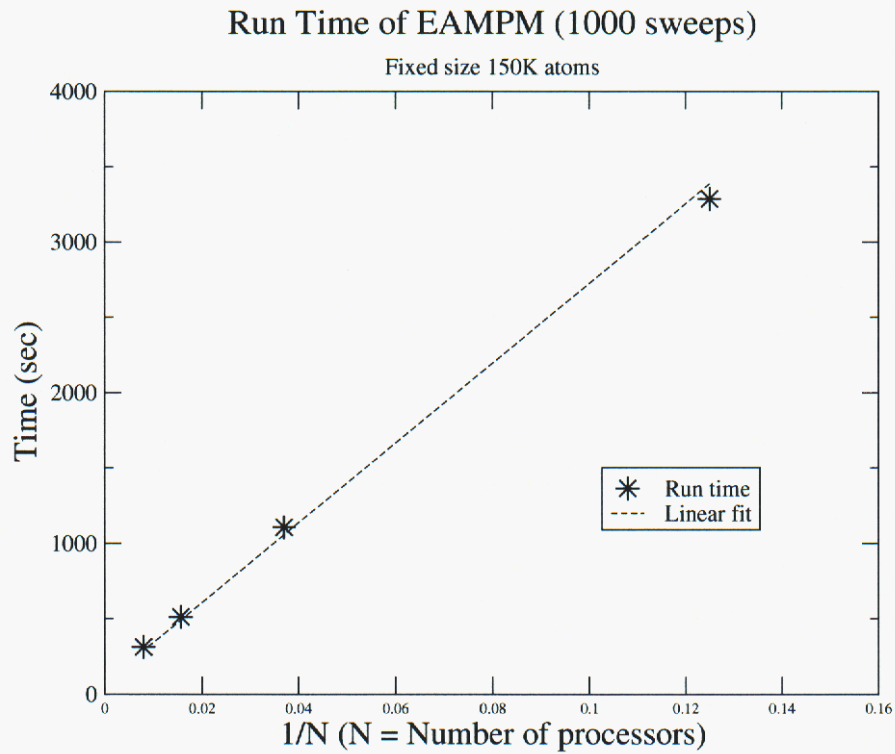
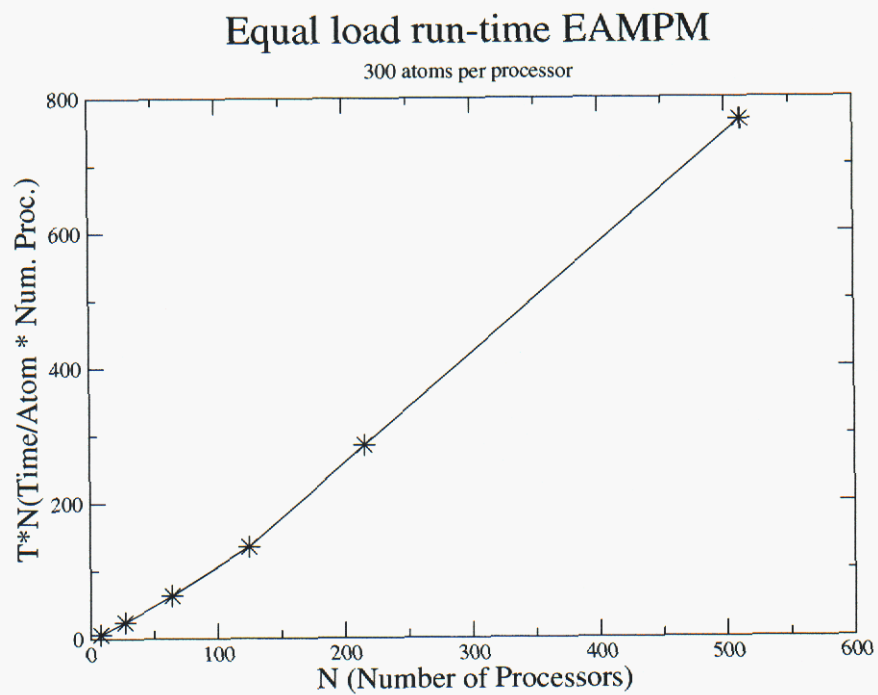


Figure 3.3: The horizontal axis is the inverse of the number of processors, while the vertical axis represents simulation time for a 1000 sweeps. Linear fit demonstrates the linear scaling of the times as a function of $1/N$.



Wed Nov 27 17:03:36 2002

Figure 3.4: The horizontal axis has the number of processors. The vertical axis represents time per number of atoms moved multiplied by the number of processors.

Chapter 4

Molecular Dynamics: the Modified Embedded Atom Method Potential in Warp

4.1 The MEAM Potential

While the embedded atom method (EAM) potential described in the previous chapter is a powerful model for atomic interactions in a bulk metal, modifications can be made to the potential make it more general. In the modified embedded atom method (MEAM) [11], additional terms are added to the expression for electron density that improve the accuracy of the model, especially near surfaces and defects where the EAM potential alone may be inadequate.

The MEAM potential has been included in the molecular dynamics code Warp, allowing the simulation of large scale MEAM systems. In this chapter, we give the form of the MEAM potential that has been implemented, along with the analytical form of the derivative of the potential used to compute the interatomic forces.

The MEAM potential has been written in various forms in different publications. In this report, the forms and notation used draws most heavily from the presentations in references [12] and [13].

The total MEAM configuration energy is a sum of direct contributions from all atoms:

$$E = \sum_i \left\{ F_i(\bar{\rho}_i) + \frac{1}{2} \sum_{i \neq j} \phi_{ij}(r_{ij}) \right\} \quad (4.1)$$

Throughout this chapter, lowercase subscripts like i and j in the above will indicate atom labels, while Greek subscripts like α and β will represent spatial directions for components of vectors and matrices. In equation (4.1), the first term corresponds to an environment-dependent energy of embedding an atom, while the second term is the pair potential between atoms.

4.1.1 Embedding energy potential

The embedding function has the form

$$F_i(\bar{\rho}_i) = A_i E_i^0 \bar{\rho}_i \ln \bar{\rho}_i \quad (4.2)$$

where sublimation energy E_i^0 and parameter A_i depend on the element type of atom i . The background electron density $\bar{\rho}_i$ is given by:

$$\bar{\rho}_i = \frac{\bar{\rho}_i^{(0)}}{\rho_i^0} G_i(\Gamma_i) \quad (4.3)$$

where

$$\Gamma_i = \sum_{k=1}^3 t_i^{(k)} \left(\frac{\bar{\rho}_i^{(k)}}{\bar{\rho}_i^{(0)}} \right)^2. \quad (4.4)$$

and $G_i(\Gamma)$ is a function that can be different for different element types in a simulation. Common choices for $G(\Gamma)$ are $G(\Gamma) = \sqrt{1 + \Gamma}$ and $G(\Gamma) = \exp(\Gamma/2)$. The zeroth and higher order densities $\bar{\rho}_i^{(k)}$ for $k = 0 - 3$ are given below in equation 4.7.

The composition-dependent electron density scaling ρ_i^0 is:

$$\rho_i^0 = \rho_{i0} Z_{i0} G_i(\Gamma_i^{ref}) \quad (4.5)$$

In this expression ρ_{i0} is an element-dependent density scaling, Z_{i0} is the first-neighbor coordination of the reference system, and Γ_i^{ref} is given by

$$\Gamma_i^{ref} = \sum_{k=1}^3 t_i^{(k)} \frac{s_i^{(k)}}{Z_{i0}^2} \quad (4.6)$$

where $s_i^{(k)}$ are shape factors that depend on the reference structure for atom i . Shape factors for various structures are given in reference [11].

The partial electron densities are given by

$$\bar{\rho}_i^{(0)} = \sum_{j \neq i} \rho_j^{a(0)}(r_{ij}) S_{ij} \quad (4.7a)$$

$$(\bar{\rho}_i^{(1)})^2 = \sum_{\alpha=1}^3 \left[\sum_{j \neq i} \rho_j^{a(1)} \frac{r_{ij\alpha}}{r_{ij}} S_{ij} \right]^2 \quad (4.7b)$$

$$(\bar{\rho}_i^{(2)})^2 = \sum_{\alpha=1}^3 \sum_{\beta=1}^3 \left[\sum_{j \neq i} \rho_j^{a(2)} \frac{r_{ij\alpha} r_{ij\beta}}{r_{ij}^2} S_{ij} \right]^2 - \frac{1}{3} \left[\sum_{j \neq i} \rho_j^{a(2)}(r_{ij}) S_{ij} \right]^2 \quad (4.7c)$$

$$(\bar{\rho}_i^{(3)})^2 = \sum_{\alpha=1}^3 \sum_{\beta=1}^3 \sum_{\gamma=1}^3 \left[\sum_{j \neq i} \rho_j^{a(3)} \frac{r_{ij\alpha} r_{ij\beta} r_{ij\gamma}}{r_{ij}^3} S_{ij} \right]^2 - \frac{3}{5} \sum_{\alpha=1}^3 \left[\sum_{j \neq i} \rho_j^{a(3)} \frac{r_{ij\alpha}}{r_{ij}} S_{ij} \right]^2 \quad (4.7d)$$

The screening function S_{ij} will be discussed in section 4.1.3. The atomic electron densities are computed as

$$\rho_i^{a(k)}(r_{ij}) = \rho_{i0} \exp \left[-\beta_i^{(k)} \left(\frac{r_{ij}}{r_i^0} - 1 \right) \right] \quad (4.8)$$

where r_i^0 is the nearest-neighbor distance in the single-element reference structure and $\beta_i^{(k)}$ are element-dependent parameters.

Finally, the average weighting factors are given by

$$t_i^{(k)} = \frac{1}{\bar{\rho}_i^{(0)}} \sum_{j \neq i} t_{0,j}^{(k)} \rho_j^{a(0)} S_{ij} \quad (4.9)$$

where $t_{0,j}^{(k)}$ are element-dependent parameters.

4.1.2 Pair potential

The pair potential, including the screening effect, is given by:

$$\phi_{ij}(r_{ij}) = \bar{\phi}_{ij}(r_{ij}) S_{ij} \quad (4.10a)$$

$$\bar{\phi}_{ij}(r_{ij}) = \frac{1}{Z_{ij0}} [2E_i^u(r_{ij}) - F_i(\hat{\rho}_i(r_{ij})) - F_j(\hat{\rho}_j(r_{ij}))] \quad (4.10b)$$

$$E_i^u(r_{ij}) = -E_{ij}^0 (1 + a_{ij}^*(r_{ij})) e^{-a_{ij}^*(r_{ij})} \quad (4.10c)$$

$$a_{ij}^* = \alpha_{ij} \left(\frac{r_{ij}}{r_{ij}^0} - 1 \right) \quad (4.10d)$$

where E_{ij}^0 , α_{ij}^0 and r_{ij}^0 are parameters that depend on the elements of i and j , while Z_{ij0} depends on the structure of the reference system. The background densities $\hat{\rho}_i(r_{ij})$ in equation (4.10b) are the densities for the reference structure computed with interatomic spacing r_{ij} .

Equation (4.10b) can have more complicated forms for some reference structures, but in general the derivative of $\bar{\phi}_{ij}$ with respect to r_{ij} is straightforward.

4.1.3 Screening function

The screening function S_{ij} is designed so that $S_{ij} = 1$ if atoms i and j are unscreened and within the cutoff radius r_c , and $S_{ij} = 0$ if they are completely screened or outside the cutoff radius. It varies smoothly between 0 and 1 for partial screening. The total screening function is the product of a radial cutoff function and three-body terms involving all other atoms in

the system:

$$S_{ij} = \bar{S}_{ij} f_c \left(\frac{r_c - r_{ij}}{\Delta r} \right) \quad (4.11a)$$

$$\bar{S}_{ij} = \prod_{k \neq i, j} S_{ikj} \quad (4.11b)$$

$$S_{ikj} = f_c \left(\frac{C_{ikj} - C_{\min, ikj}}{C_{\max, ikj} - C_{\min, ikj}} \right) \quad (4.11c)$$

$$C_{ikj} = 1 + 2 \frac{r_{ij}^2 r_{ik}^2 + r_{ij}^2 r_{jk}^2 - r_{ij}^4}{r_{ij}^4 - (r_{ik}^2 - r_{jk}^2)^2} \quad (4.11d)$$

$$f_c(x) = \begin{cases} 1 & x \geq 1 \\ [1 - (1 - x)^4]^2 & 0 < x < 1 \\ 0 & x \leq 0 \end{cases} \quad (4.11e)$$

Note that C_{\min} and C_{\max} can be defined separately for each $i - j - k$ triplet, based on their element types. The parameter Δr controls the distance over which the radial cutoff is smoothed from 1 to 0 near $r = r_c$.

4.2 Notes on Notation

Throughout this chapter, we will use a comma/subscript notation to denote derivatives. A lowercase letter and a Greek letter after a comma in a subscript denote differentiation with respect to the position of a single atom:

$$f_{,i\alpha} \equiv \frac{\partial f}{\partial x_{i\alpha}} \quad (4.12)$$

Two lowercase letters after a comma denote differentiation with respect to the distance between two atoms:

$$f_{,ij} \equiv \frac{\partial f}{\partial r_{ij}} \quad (4.13)$$

Except where indicated, two lowercase letters plus a Greek letter after a comma denote differentiation with respect to a single component of the vector between two atoms:

$$f_{,ij\alpha} \equiv \frac{\partial f}{\partial r_{ij\alpha}} \quad (4.14)$$

where

$$r_{ij\alpha} = x_{j\alpha} - x_{i\alpha} \quad (4.15)$$

Note, then, that

$$r_{ij\alpha} = -r_{ji\alpha} \quad (4.16a)$$

$$f_{,ij\alpha} = -f_{,ji\alpha} \quad (4.16b)$$

4.3 Derivatives of the Screening Function

The screening function S_{ij} is a function only of the distances between atoms. We will need only to consider only three derivatives of S_{ij} : that with respect to the distance between the two screened atoms, r_{ij} , and those with respect to the distance between either of the screened atoms and a third (possibly screening) atom, r_{ik} and r_{jk} .

First we differentiate the parameter C_{ikj} , and introduce some notation along the way:

$$\partial C_{ikj}^{\textcircled{0}} \equiv C_{ikj,ij} = -\frac{4r_{ij}(2r_{ij}^2(r_{ik}^2 - r_{jk}^2)^2 + (r_{ij}^4 + (r_{ik}^2 - r_{jk}^2)^2)(r_{ik}^2 + r_{jk}^2))}{(r_{ij}^4 - (r_{ij}^2 - r_{jk}^2)^2)^2} \quad (4.17a)$$

$$\partial C_{ikj}^{\textcircled{1}} \equiv C_{ikj,ik} = \frac{4r_{ij}^2 r_{ik}(r_{ij}^4 + r_{ik}^4 + 2r_{ik}^2 r_{jk}^2 - 3r_{jk}^4 - 2r_{ij}^2(r_{ik}^2 - r_{jk}^2))}{(r_{ij}^4 - (r_{ij}^2 - r_{jk}^2)^2)^2} \quad (4.17b)$$

$$\partial C_{ikj}^{\textcircled{2}} \equiv C_{ikj,jk} = \frac{4r_{ij}^2 r_{jk}(r_{ij}^4 + r_{jk}^4 + 2r_{ik}^2 r_{jk}^2 - 3r_{ik}^4 + 2r_{ij}^2(r_{ik}^2 - r_{jk}^2))}{(r_{ij}^4 - (r_{ij}^2 - r_{jk}^2)^2)^2} \quad (4.17c)$$

Note that, because C_{ikj} is symmetric in $i - j$:

$$\partial C_{jki}^{\textcircled{0}} = \partial C_{ikj}^{\textcircled{0}} \quad (4.18a)$$

$$\partial C_{jki}^{\textcircled{1}} = \partial C_{ikj}^{\textcircled{2}} \quad (4.18b)$$

For the 3-body screening term S_{ikj} , we can write for a general r_{lm}

$$S_{ikj,lm} = \frac{1}{\Delta C_{ikj}} f'_{C_{ikj}} C_{ikj,lm} \quad (4.19)$$

where

$$\Delta C_{ikj} \equiv C_{\max,ikj} - C_{\min,ikj} \quad (4.20a)$$

$$f'_{C_{ikj}} \equiv f'_c \left(\frac{C_{ikj} - C_{\min,ikj}}{C_{\max,ikj} - C_{\min,ikj}} \right) \quad (4.20b)$$

The derivative of \bar{S}_{ij} , the product of these three-body terms, is

$$\begin{aligned} \bar{S}_{ij,lm} &= \sum_{k \neq i,j} \left(S_{ikj,lm} \prod_{n \neq i,j,k} S_{inj} \right) \\ &= \sum_{k \neq i,j} S_{ikj,lm} \frac{1}{\bar{S}_{ikj}} \prod_{n \neq i,j} S_{inj} \\ &= \bar{S}_{ij} \sum_{k \neq i,j} \frac{1}{S_{ikj}} S_{ikj,lm} \\ &= \bar{S}_{ij} \sum_{k \neq i,j} \frac{1}{S_{ikj} \Delta C_{ikj}} f'_{C_{ikj}} C_{ikj,lm} \end{aligned} \quad (4.21)$$

Finally, we can include the effects of taking the product with the radial cutoff function (which only contributes to the r_{ij} derivative), and use the derivatives of C_{ikj} given in (4.17). At the same time, we introduce a notation for the derivatives of S_{ij} :

$$\partial S_{ij}^{\textcircled{0}} \equiv S_{ij,ij} = f_{r_{ij}} \bar{S}_{ij} \sum_{k \neq i,j} \frac{1}{\Delta C_{ikj} S_{ikj}} f'_{C_{ikj}} \partial C_{ikj}^{\textcircled{0}} - \frac{1}{\Delta r} \bar{S}_{ij} f'_{r_{ij}} \quad (4.22a)$$

$$\partial S_{ij,k}^{\textcircled{1}} \equiv S_{ij,ik} = f_{r_{ij}} \bar{S}_{ij} \frac{1}{\Delta C_{ikj} S_{ikj}} f'_{C_{ikj}} \partial C_{ikj}^{\textcircled{1}} \quad (4.22b)$$

$$\partial S_{ij,k}^{\textcircled{2}} \equiv S_{ij,ik} = f_{r_{ij}} \bar{S}_{ij} \frac{1}{\Delta C_{ikj} S_{ikj}} f'_{C_{ikj}} \partial C_{ikj}^{\textcircled{2}} \quad (4.22c)$$

where

$$f_{r_{ij}} \equiv f_c \left(\frac{r_c - r_{ij}}{\Delta r} \right) \quad (4.23a)$$

$$f'_{r_{ij}} \equiv f'_c \left(\frac{r_c - r_{ij}}{\Delta r} \right) \quad (4.23b)$$

Equations (4.17) and (4.22) are the main results of this section.

With the notation defined in equation (4.22), we can write the derivative of the screening function S_{ij} with respect to an arbitrary pair distance r_{lm} . We apply the chain rule, treating in turn the dependence of S_{ij} on r_{ij} , r_{ik} , and r_{jk} :

$$\begin{aligned} \frac{\partial S_{ij}}{\partial r_{lm}} &= \frac{\partial S_{ij}}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial r_{lm}} + \sum_{k \neq i,j} \left(\frac{\partial S_{ij}}{\partial r_{ik}} \frac{\partial r_{ik}}{\partial r_{lm}} + \frac{\partial S_{ij}}{\partial r_{jk}} \frac{\partial r_{jk}}{\partial r_{lm}} \right) \\ &= \partial S_{ij}^{\textcircled{0}} (\delta_{il} \delta_{jm} + \delta_{im} \delta_{jl}) + \sum_{k \neq i,j} \left[\partial S_{ij,k}^{\textcircled{1}} (\delta_{il} \delta_{km} + \delta_{im} \delta_{kl}) + \partial S_{ij,k}^{\textcircled{2}} (\delta_{jl} \delta_{km} + \delta_{jm} \delta_{kl}) \right] \\ &= \partial S_{ij}^{\textcircled{0}} (\delta_{il} \delta_{jm} + \delta_{im} \delta_{jl}) + \partial S_{ij,m}^{\textcircled{1}} \delta_{il} + \partial S_{ij,l}^{\textcircled{1}} \delta_{im} + \partial S_{ij,m}^{\textcircled{2}} \delta_{jl} + \partial S_{ij,l}^{\textcircled{2}} \delta_{jm} \end{aligned} \quad (4.24)$$

4.4 Derivatives of the Electron Densities

Looking ahead to section 4.5, we will require the derivatives of the electron densities with respect to interatomic distances r_{lm} , screening functions S_{lm} , and individual spatial components of interatomic distances $r_{lm\alpha}$. Here, we are interested in the *explicit* dependence of the densities on these arguments; thus, for example, a function that depends on r_{lm} alone has zero derivative with respect to $r_{lm\alpha}$, even though r_{lm} itself is implicitly a function of $r_{lm\alpha}$. Likewise, the implicit dependence of S_{lm} on r_{lm} is ignored here.

In this section, we give these derivatives, along with some new notation that takes advantage of the convenient forms of some of the derivatives. We begin by finding the derivatives of the zeroth and higher order electron densities, and then build the derivatives of the total background density out of these results.

4.4.1 Zeroth-order density

The important derivatives of $\bar{\rho}_i^{(0)}$ (equation 4.7a) have the forms

$$\frac{\partial \bar{\rho}_i^{(0)}}{\partial r_{lm}} = \partial \bar{\rho}_{im}^{(0)\textcircled{\tiny R}} \delta_{il} + \partial \bar{\rho}_{il}^{(0)\textcircled{\tiny R}} \delta_{im} \quad (4.25a)$$

$$\frac{\partial \bar{\rho}_i^{(0)}}{\partial S_{lm}} = \partial \bar{\rho}_{im}^{(0)\textcircled{\tiny S}} \delta_{il} + \partial \bar{\rho}_{il}^{(0)\textcircled{\tiny S}} \delta_{im} \quad (4.25b)$$

$$\frac{\partial \bar{\rho}_i^{(0)}}{\partial r_{lm\alpha}} = 0 \quad (4.25c)$$

where δ_{il} and δ_{im} are kronecker deltas, and

$$\partial \bar{\rho}_{ij}^{(0)\textcircled{\tiny R}} \equiv \rho_j^{a(0)'}(r_{ij}) S_{ij} \quad (4.26a)$$

$$\partial \bar{\rho}_{ij}^{(0)\textcircled{\tiny S}} \equiv \rho_j^{a(0)}(r_{ij}) \quad (4.26b)$$

4.4.2 Higher-order densities

We will first introduce notation to simplify the nested summations required for computation of the higher-order densities. Define:

$$Y_{1i\sigma} = \sum_{j \neq i} \rho_j^{a(1)} \frac{r_{ij\sigma}}{r_{ij}} S_{ij} \quad (4.27a)$$

$$Y_{2i\sigma\beta} = \sum_{j \neq i} \rho_j^{a(2)} \frac{r_{ij\sigma} r_{ij\beta}}{r_{ij}^2} S_{ij} \quad (4.27b)$$

$$Y_{3i\sigma\beta\gamma} = \sum_{j \neq i} \rho_j^{a(3)} \frac{r_{ij\sigma} r_{ij\beta} r_{ij\gamma}}{r_{ij}^3} S_{ij} \quad (4.27c)$$

$$W_{2i} = \sum_{j \neq i} \rho_j^{a(2)} S_{ij} \quad (4.27d)$$

$$W_{3i\sigma} = \sum_{j \neq i} \rho_j^{a(3)} \frac{r_{ij\sigma}}{r_{ij}} S_{ij} \quad (4.27e)$$

Then the higher order densities (4.7) can be written:

$$(\bar{\rho}_i^{(1)})^2 = \sum_{\sigma=1}^3 Y_{1i\sigma}^2 \quad (4.28a)$$

$$(\bar{\rho}_i^{(2)})^2 = \sum_{\sigma=1}^3 \sum_{\beta=1}^3 Y_{2i\sigma\beta}^2 - \frac{1}{3} W_{2i}^2 \quad (4.28b)$$

$$(\bar{\rho}_i^{(3)})^2 = \sum_{\sigma=1}^3 \sum_{\beta=1}^3 \sum_{\gamma=1}^3 Y_{3i\sigma\beta\gamma}^2 - \frac{3}{5} \sum_{\sigma=1}^3 W_{3i\sigma}^2 \quad (4.28c)$$

The derivatives of $(\bar{\rho}_i^{(k)})^2$ have the forms

$$\frac{\partial(\bar{\rho}_i^{(k)})^2}{\partial r_{lm}} = \partial\bar{\rho}_{im}^{(k)\textcircled{R}}\delta_{il} + \partial\bar{\rho}_{il}^{(k)\textcircled{R}}\delta_{im} \quad (4.29a)$$

$$\frac{\partial(\bar{\rho}_i^{(k)})^2}{\partial S_{lm}} = \partial\bar{\rho}_{im}^{(k)\textcircled{S}}\delta_{il} + \partial\bar{\rho}_{il}^{(k)\textcircled{S}}\delta_{im} \quad (4.29b)$$

$$\frac{\partial(\bar{\rho}_i^{(0)})^2}{\partial r_{lm\alpha}} = \partial\bar{\rho}_{im\alpha}^{(k)}\delta_{il} - \partial\bar{\rho}_{il\alpha}^{(k)}\delta_{im} \quad (4.29c)$$

(note the minus sign in (4.29c), due to the antisymmetry of $r_{lm\alpha}$). The individual terms in these derivatives are computed as given below:

$$\partial\bar{\rho}_{ij}^{(1)\textcircled{R}} = \frac{2}{r_{ij}}S_{ij}\left(\rho_j^{a(1)'}(r_{ij}) - \frac{1}{r_{ij}}\rho_j^{a(1)}(r_{ij})\right)\left(\sum_{\sigma}Y_{1i\sigma}r_{ij\sigma}\right) \quad (4.30a)$$

$$\partial\bar{\rho}_{ij}^{(1)\textcircled{S}} = \frac{2}{r_{ij}}\rho_j^{a(1)}(r_{ij})\sum_{\sigma}Y_{1i\sigma}r_{ij\sigma} \quad (4.30b)$$

$$\partial\bar{\rho}_{ij\alpha}^{(1)} = \frac{2}{r_{ij}}S_{ij}Y_{1i\alpha}\rho_j^{a(1)}(r_{ij}) \quad (4.30c)$$

$$\begin{aligned} \partial\bar{\rho}_{ij}^{(2)\textcircled{R}} = \frac{2}{r_{ij}^2}S_{ij}\left(\rho_j^{a(2)'}(r_{ij}) - \frac{2}{r_{ij}}\rho_j^{a(2)}(r_{ij})\right)\left(\sum_{\sigma}\sum_{\beta}Y_{2i\sigma\beta}r_{ij\sigma}r_{ij\beta}\right) \\ - \frac{2}{3}W_{2i}\rho_j^{a(2)'}(r_{ij})S_{ij} \end{aligned} \quad (4.30d)$$

$$\partial\bar{\rho}_{ij}^{(2)\textcircled{S}} = \frac{2}{r_{ij}^2}\rho_j^{a(2)}(r_{ij})\left(\sum_{\sigma}\sum_{\beta}Y_{2i\sigma\beta}r_{ij\sigma}r_{ij\beta}\right) - \frac{2}{3}W_{3i}\rho_j^{a(2)}(r_{ij}) \quad (4.30e)$$

$$\partial\bar{\rho}_{ij\alpha}^{(2)} = \frac{4}{r_{ij}^2}\rho_j^{a(2)}(r_{ij})S_{ij}\sum_{\sigma}Y_{2i\sigma\alpha}r_{ij\sigma} \quad (4.30f)$$

$$\begin{aligned} \partial\bar{\rho}_{ij}^{(3)\textcircled{R}} = \frac{2}{r_{ij}^3}S_{ij}\left(\rho_j^{a(3)'}(r_{ij}) - \frac{3}{r_{ij}}\rho_j^{a(3)}(r_{ij})\right)\left(\sum_{\sigma}\sum_{\beta}\sum_{\gamma}Y_{3i\sigma\beta\gamma}r_{ij\sigma}r_{ij\beta}r_{ij\gamma}\right) \\ - \frac{6}{5r_{ij}}S_{ij}\left(\rho_j^{a(3)'}(r_{ij}) - \frac{1}{r_{ij}}\rho_j^{a(3)}(r_{ij})\right)\left(\sum_{\sigma}W_{3i\sigma}r_{ij\sigma}\right) \end{aligned} \quad (4.30g)$$

$$\begin{aligned} \partial \bar{\rho}_{ij}^{(3)\textcircled{S}} = \frac{2}{r_{ij}^3} \rho_j^{a(3)}(r_{ij}) \sum_{\sigma} \sum_{\beta} \sum_{\gamma} Y_{i\sigma\beta\gamma} r_{ij\sigma} r_{ij\beta} r_{ij\gamma} \\ - \frac{6}{5r_{ij}} \rho_j^{a(3)}(r_{ij}) \sum_{\sigma} W_{3i\sigma} r_{ij\sigma} \end{aligned} \quad (4.30h)$$

$$\partial \bar{\rho}_{ij\alpha}^{(3)} = \frac{6}{r_{ij}^3} \rho_j^{a(3)}(r_{ij}) S_{ij} \sum_{\sigma} \sum_{\beta} Y_{3i\alpha\sigma\beta} r_{ij\sigma} r_{ij\beta} - \frac{6}{5r_{ij}} \rho_j^{a(3)}(r_{ij}) S_{ij} W_{3i\alpha} \quad (4.30i)$$

All summations in the above expressions are over spatial indices ranging from 1 to 3. Note that we can take advantage of the symmetry of the arguments of the nested summations; thus, double sums can be evaluated using 6 terms rather than 9, while triple sums can be evaluated using 10 terms rather than 27. This is also true of the summations in equation (4.27).

4.4.3 Derivatives of the averaged weighting factors

The derivatives of the average weighting factors $t_i^{(k)}$ (equation 4.9) have the forms

$$\frac{\partial t_i^{(k)}}{\partial r_{lm}} = \partial t_{im}^{(k)\textcircled{R}} \delta_{il} + \partial t_{il}^{(k)\textcircled{R}} \delta_{im} \quad (4.31a)$$

$$\frac{\partial t_i^{(k)}}{\partial S_{ij}} = \partial t_{im}^{(k)\textcircled{S}} \delta_{il} + \partial t_{il}^{(k)\textcircled{S}} \delta_{im} \quad (4.31b)$$

$$\frac{\partial t_i^{(k)}}{\partial r_{lm\alpha}} = 0 \quad (4.31c)$$

where

$$\partial t_{ij}^{(k)\textcircled{R}} = \frac{1}{\bar{\rho}_i^{(0)}} \left(t_{0,j}^{(k)} - t_i^{(k)} \right) \rho_j^{a(0)'}(r_{ij}) S_{ij} \quad (4.32a)$$

$$\partial t_{ij}^{(k)\textcircled{S}} = \frac{1}{\bar{\rho}_i^{(0)}} \left(t_{0,j}^{(k)} - t_i^{(k)} \right) \rho_j^{a(0)}(r_{ij}) \quad (4.32b)$$

Note as a check that for the case where all atoms are of the same element type, and therefore have the same weighting factors, $t_i^{(k)} = t_{0,j}^{(k)}$ and the derivatives are zero, as expected.

4.4.4 Derivatives of Γ_i

The derivatives of Γ_i (equation 4.4) have the forms

$$\frac{\partial \Gamma_i^{(k)}}{\partial r_{lm}} = \partial \Gamma_{im}^{\textcircled{R}} \delta_{il} + \partial \Gamma_{il}^{\textcircled{R}} \delta_{im} \quad (4.33a)$$

$$\frac{\partial \Gamma_i^{(k)}}{\partial S_{ij}} = \partial \Gamma_{im}^{\textcircled{S}} \delta_{il} + \partial \Gamma_{il}^{\textcircled{S}} \delta_{im} \quad (4.33b)$$

$$\frac{\partial \Gamma_i^{(k)}}{\partial r_{lm\alpha}} = \partial \Gamma_{im\alpha} \delta_{il} + \partial \Gamma_{il\alpha} \delta_{im} \quad (4.33c)$$

where

$$\begin{aligned} \partial \Gamma_{ij}^{\textcircled{R}} = & -\frac{2}{\bar{\rho}_i^{(0)}} \Gamma_i \partial \bar{\rho}_{ij}^{(0)\textcircled{R}} + \frac{\rho_j^{a(0)'}(r_{ij}) S_{ij}}{(\bar{\rho}_i^{(0)})^3} \sum_{k=1}^3 \left(t_{0,j}^{(k)} - t_i^{(k)} \right) (\bar{\rho}_i^{(k)})^2 \\ & + \frac{1}{(\bar{\rho}_i^{(0)})^2} \sum_{k=1}^3 t_i^{(k)} \partial \bar{\rho}_{ij}^{(k)\textcircled{R}} \end{aligned} \quad (4.34a)$$

$$\begin{aligned} \partial \Gamma_{ij}^{\textcircled{S}} = & -\frac{2}{\bar{\rho}_i^{(0)}} \Gamma_i \partial \bar{\rho}_{ij}^{(0)\textcircled{S}} + \frac{\rho_j^{a(0)'}(r_{ij})}{(\bar{\rho}_i^{(0)})^3} \sum_{k=1}^3 \left(t_{0,j}^{(k)} - t_i^{(k)} \right) (\bar{\rho}_i^{(k)})^2 \\ & + \frac{1}{(\bar{\rho}_i^{(0)})^2} \sum_{k=1}^3 t_i^{(k)} \partial \bar{\rho}_{ij}^{(k)\textcircled{S}} \end{aligned} \quad (4.34b)$$

$$\partial \Gamma_{ij\alpha} = \frac{1}{(\bar{\rho}_i^{(0)})^2} \sum_{k=1}^3 t_i^{(k)} \partial \bar{\rho}_{ij}^{(k)} \quad (4.34c)$$

4.4.5 Derivative of the total density

The derivative of the total background density (equation 4.3) can be easily derived using the derivatives in the previous subsections. They take the forms:

$$\frac{\partial \bar{\rho}_i^{(k)}}{\partial r_{lm}} = \partial \bar{\rho}_{im}^{\textcircled{R}} \delta_{il} + \partial \bar{\rho}_{il}^{\textcircled{R}} \delta_{im} \quad (4.35a)$$

$$\frac{\partial \bar{\rho}_i^{(k)}}{\partial S_{ij}} = \partial \bar{\rho}_{im}^{\textcircled{S}} \delta_{il} + \partial \bar{\rho}_{il}^{\textcircled{S}} \delta_{im} \quad (4.35b)$$

$$\frac{\partial \bar{\rho}_i^{(k)}}{\partial r_{lm\alpha}} = \partial \bar{\rho}_{im\alpha} \delta_{il} + \partial \bar{\rho}_{il\alpha} \delta_{im} \quad (4.35c)$$

where

$$\begin{aligned} \partial \bar{\rho}_{ij}^{\textcircled{\Gamma}} = \frac{1}{\bar{\rho}_i^0} & \left[(G_i(\Gamma_i) - 2G'_i(\Gamma_i)\Gamma_i) \partial \bar{\rho}_{ij}^{(0)\textcircled{\Gamma}} \right. \\ & + \frac{1}{\bar{\rho}_i^{(0)}} G'_i(\Gamma_i) \sum_{k=1}^3 \left[\partial t_{ij}^{(k)\textcircled{\Gamma}} (\bar{\rho}_i^{(k)})^2 + t_i^{(k)} \partial \bar{\rho}_{ij}^{(k)\textcircled{\Gamma}} \right] \\ & \left. - \frac{\bar{\rho}_i \rho_{i0}}{Z_{i0}} G'_i(\Gamma_i^{ref}) \sum_{k=1}^3 s_i^{(k)} \partial t_{ij}^{(k)\textcircled{\Gamma}} \right] \end{aligned} \quad (4.36a)$$

$$\begin{aligned} \partial \bar{\rho}_{ij}^{\textcircled{\mathbb{S}}} = \frac{1}{\bar{\rho}_i^0} & \left[(G_i(\Gamma_i) - 2G'_i(\Gamma_i)\Gamma_i) \partial \bar{\rho}_{ij}^{(0)\textcircled{\mathbb{S}}} \right. \\ & + \frac{1}{\bar{\rho}_i^{(0)}} G'_i(\Gamma_i) \sum_{k=1}^3 \left[\partial t_{ij}^{(k)\textcircled{\mathbb{S}}} (\bar{\rho}_i^{(k)})^2 + t_i^{(k)} \partial \bar{\rho}_{ij}^{(k)\textcircled{\mathbb{S}}} \right] \\ & \left. - \frac{\bar{\rho}_i \rho_{i0}}{Z_{i0}} G'_i(\Gamma_i^{ref}) \sum_{k=1}^3 s_i^{(k)} \partial t_{ij}^{(k)\textcircled{\mathbb{S}}} \right] \end{aligned} \quad (4.36b)$$

$$\partial \bar{\rho}_{ij\alpha} = \frac{1}{\bar{\rho}_i^0 \bar{\rho}_i^{(0)}} G'_i(\Gamma_i) \sum_{k=1}^3 t_i^{(k)} \partial \bar{\rho}_{ij\alpha}^{(k)} \quad (4.36c)$$

Note that $\bar{\rho}_i^0$ in these expressions is the composition-dependent electron density scaling given in equation (4.5).

Equations (4.35) and (4.36), together with the derivatives of the higher-order densities in equation (4.30), are the main results of this section.

4.5 Derivatives of the MEAM Energy

With the derivatives of the electron density in hand, we can compute the derivative of the full MEAM potential energy. We have a choice between computing the derivatives with respect to individual atomic positions $x_{i\alpha}$, or to pairwise distances $r_{ij\alpha}$. The force on an atom is most directly computed from the former:

$$f_{i\alpha} = -\frac{\partial E}{\partial x_{i\alpha}} \quad (4.37)$$

However, by the chain rule this can be converted to a form involving the pair derivatives:

$$\begin{aligned}
 f_{i\alpha} &= - \sum_{j \neq i} \sum_{\beta=1}^3 \frac{\partial E}{\partial r_{ij\beta}} \frac{\partial r_{ij\beta}}{\partial x_{i\alpha}} \\
 &= \sum_{j \neq i} \frac{\partial E}{\partial r_{ij\alpha}} \\
 &= \sum_{j \neq i} f_{ij\alpha}
 \end{aligned} \tag{4.38}$$

where

$$f_{ij\alpha} \equiv \frac{\partial E}{\partial r_{ij\alpha}} \tag{4.39}$$

The chain rule can be used to write this derivative in terms of the explicit dependencies on r_{ij} , S_{ij} , and $r_{ij\alpha}$. Note that many of the pairwise screening functions (and not just that between atoms i and j) can depend on $r_{ij\alpha}$, and thus we must take into account all of these terms in the chain rule and use equation 4.24 to simplify the expression.

$$\begin{aligned}
 \frac{\partial E}{\partial r_{ij\alpha}} &= \frac{\partial E}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial r_{ij\alpha}} + \frac{1}{2} \sum_l \sum_{m \neq l} \frac{\partial E}{\partial S_{lm}} \frac{\partial S_{lm}}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial r_{ij\alpha}} + \frac{\partial E}{\partial r_{ij\alpha}} \\
 &= \left[\frac{\partial E}{\partial r_{ij}} + \frac{\partial E}{\partial S_{ij}} \partial S_{ij}^{\textcircled{0}} + \sum_{k \neq i,j} \left(\frac{\partial E}{\partial S_{ik}} \partial S_{ik}^{\textcircled{1}} + \frac{\partial E}{\partial S_{jk}} \partial S_{jk}^{\textcircled{1}} \right) \right] \frac{r_{ij\alpha}}{r_{ij}} + \frac{\partial E}{\partial r_{ij\alpha}}
 \end{aligned} \tag{4.40}$$

The coefficient 1/2 in the first line is due to the symmetry of S_{lm} .

It should be understood that the $r_{ij\alpha}$ derivative on the left hand side of 4.40, and in the definition of equation 4.39, refers to all of the dependencies (both explicit and implicit) of E on $r_{ij\alpha}$, while that on the right hand side of 4.40 includes only the explicit dependencies (i.e. holding all r_{ij} and S_{ij} constant). Also, note that all nested summations over atoms are restricted to unlike pairs, even where not noted; thus, the summations over l and m in the above expression would more accurately (though less compactly) be written $\sum_{l \neq i,j}$ and $\sum_{m \neq l,i,j}$.

The individual derivatives of the energy are:

$$\frac{\partial E}{\partial r_{ij}} = F'_i(\bar{\rho}_i) \partial \bar{\rho}_{ij}^{\textcircled{\mathbf{r}}} + F'_j(\bar{\rho}_j) \partial \bar{\rho}_{ji}^{\textcircled{\mathbf{r}}} + \bar{\phi}'_{ij}(r_{ij}) S_{ij} \tag{4.41a}$$

$$\frac{\partial E}{\partial S_{ij}} = F'_i(\bar{\rho}_i) \partial \bar{\rho}_{ij}^{\textcircled{\mathbf{S}}} + F'_j(\bar{\rho}_j) \partial \bar{\rho}_{ji}^{\textcircled{\mathbf{S}}} + \bar{\phi}_{ij}(r_{ij}) \tag{4.41b}$$

$$\frac{\partial E}{\partial r_{ij\alpha}} = F'_i(\bar{\rho}_i) \partial \bar{\rho}_{ij\alpha} + F'_j(\bar{\rho}_j) \partial \bar{\rho}_{ji\alpha} \tag{4.41c}$$

$$\tag{4.41d}$$

where $\partial \bar{\rho}_{ij}^{\textcircled{\text{T}}}$, $\partial \bar{\rho}_{ij}^{\textcircled{\text{S}}}$ and $\partial \bar{\rho}_{ij\alpha}$ are given in (4.36). The derivative of the embedding function, from (4.2), is

$$F'_i(\bar{\rho}_i) = A_i E_i^0 (\ln \bar{\rho}_i + 1). \quad (4.42)$$

The derivative of the pair potential function $\bar{\phi}_{ij}$ depends on the reference structure. In the current implementation of the MEAM potential, the pair potential functions for all element pairs is tabulated for discrete values of r_{ij} before the simulation begins. The function and its derivative are then computed from this table using cubic spline interpolation during the simulation.

Chapter 5

Postprocessing

5.1 Introduction

This chapter presents a method for computing strain tensors from the incremental motions generated by atomistic simulations that are consistent with strain tensors defined in the continuum mechanics framework. Strain tensors are the primary measure of local distortions used in continuum analysis. Unlike displacement, strain is not a measurable quantity, but is computed from a definition that relies on the gradient of a continuous displacement field. Atomistic simulations provide detailed information regarding atomic displacements; however, the discrete form of the data makes the definition of local distortion in terms of strain troublesome, and thus correlation with larger scale continuum results can be difficult.

In order to provide a common measure of distortion, a local atomic strain tensor needs to be defined. Because strain is fundamentally a continuum quantity, its computation from atomic displacement data requires either interpolation of a continuous displacement field or a discretization of the gradient operator. In this work we present the latter approach.

At least two methods for computation of a strain tensor have been proposed in the literature. Mott [14] presents a local atomic strain measure for three-dimensional, disordered systems such as glass. Here, the atomic displacements are interpolated using a continuous, piecewise-linear basis formed by a Delaunay tessellation of the atomic positions. The displacement at any point in a tetrahedron is a linear function of the displacements of the atoms at its vertices, and the deformation gradient is constant. The local atomic deformation gradient is then defined as a weighted average of the deformation gradients of adjacent tetrahedrons. This approach can be computationally expensive due to the required geometric decomposition, thus its usefulness for simulations that include significant plastic flow is limited.

Falk [15] constructs a local atomic strain tensor using small-strain definitions that do not directly depend on the deformation gradient. In this approach, a locally constant atomic strain tensor is computed based on the relative motion of an atom and its nearest neighbors. Although this formulation is straightforward, its application is restricted because a small-strain formulation does not account for strain history and is only accurate for very small

strains.

We propose an atomic strain tensor that is appropriate for finite, multi-axial deformation states, that accounts for interactions between many atoms in a nonlocal fashion, and is based on the definition of a discrete equivalent to the continuum deformation gradient.

Notation is standard, with Cartesian tensors being distinguished from scalars by bold font. The term discrete should be understood to refer spatially discontinuous nature of a body composed of a finite number of atoms, and incremental refers to the temporally discrete sequence of atomic configurations generated by atomistic simulations.

5.2 Strain Formulation

In this section we briefly consider the continuum deformation gradient, introduce a discrete version of the deformation gradient, and finally present the strain tensors.

5.2.1 Continuum Deformation Gradient

As motivation for a discrete deformation gradient, we first consider some fundamental aspects of continuum mechanics. For simplicity, assume a fixed Cartesian coordinate system and assume the particles are identified by the position vector that specify their place in the initial configuration. Now consider a motion χ that maps a particle from its initial position \mathbf{X} in the reference configuration Ω_0 , to its position \mathbf{x} in the current configuration Ω_1 ,

$$\mathbf{x} = \chi(\mathbf{X}). \quad (5.1)$$

Assuming sufficient continuity, the local deformation is characterized as the gradient of the motion, which is a second order, two-point tensor defined as

$$\mathbf{F} \equiv \frac{\partial \chi}{\partial \mathbf{X}} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}. \quad (5.2)$$

The deformation of an infinitesimal segment $d\mathbf{X}$ at point \mathbf{X} in the reference configuration is given by:

$$d\mathbf{x} = \chi(\mathbf{X} + d\mathbf{X}) - \chi(\mathbf{X}). \quad (5.3)$$

Expanding 5.3 in a Taylor series,

$$d\mathbf{x} = \chi(\mathbf{X}) + \nabla \chi(\mathbf{X}) \cdot d\mathbf{X} + o(d\mathbf{X}) - \chi(\mathbf{X}) \quad (5.4)$$

$$d\mathbf{x} \approx \nabla \chi(\mathbf{X}) \cdot d\mathbf{X}. \quad (5.5)$$

In 5.4, the notation $o(\cdot)$ takes the usual meaning of a term that approach zero faster than its argument (\cdot) . Substituting \mathbf{F} from 5.2 into 5.5 results in:

$$d\mathbf{x} \approx \mathbf{F} \cdot d\mathbf{X}. \quad (5.6)$$

Similarly, because of regularity assumptions the reverse mapping is given by

$$\mathbf{F}^{-1}d\mathbf{x} \approx d\mathbf{X}. \quad (5.7)$$

Thus \mathbf{F} maps an arbitrary, infinitesimal vector $d\mathbf{X}$ at \mathbf{X} in the reference configuration to $d\mathbf{x}$ at \mathbf{x} in the current configuration.

5.2.2 Strain Tensors

Strain is a measure of the distortion of the vector as a result of the mapping between configurations. While many strain measures are possible, e.g. see [16], we consider two common measures. From equation 5.6, the change in the squared length of the vector $d\mathbf{X}$ is given by

$$d\mathbf{x} \cdot d\mathbf{x} - d\mathbf{X} \cdot d\mathbf{X} = d\mathbf{X} \cdot \mathbf{F}^T \mathbf{F} d\mathbf{X} - d\mathbf{X} \cdot d\mathbf{X}, \quad (5.8)$$

or

$$d\mathbf{x} \cdot d\mathbf{x} - d\mathbf{X} \cdot d\mathbf{X} = d\mathbf{X} \cdot (\mathbf{F}^T \mathbf{F} - \mathbf{I}) d\mathbf{X}. \quad (5.9)$$

Here the distortion is characterized entirely in terms of the reference configuration by the dimensionless, symmetric second order tensor

$$\mathbf{F}^T \mathbf{F} - \mathbf{I}. \quad (5.10)$$

The Lagrangian or Green strain tensor, \mathbf{C} , is defined with respect to reference coordinates in terms of this quantity as

$$\mathbf{C} = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I}). \quad (5.11)$$

The Eulerian or Almansi strain tensor, \mathbf{B} , which described in terms of the current configuration is derived from equation 5.7 and is given by

$$\mathbf{B} = \frac{1}{2} (\mathbf{I} - (\mathbf{F} \mathbf{F}^T)^{-1}). \quad (5.12)$$

At small strains, when the deformation gradient components are small compared to \mathbf{I} , these measures are equivalent. The Green strain is more suitably used in elasticity, since there is usually an undeformed state to which the body is elastically unloaded. The Almansi strain is more suitably used for large strain measures, because of the presence of dislocations in the distorted the lattice.

5.2.3 Discrete Deformation Gradient

The absence of a continuous displacement field prevents computation of the deformation gradient (equation 5.2) from atomistic simulation data. In this subsection, we propose a deformation gradient that emerges from a optimization of a discrete form of equation 5.6.

A few definitions are now introduced. The generic time increment under consideration is assumed to be such that $t \in [t_0, t_{n+1}]$. The atomic configurations under consideration at

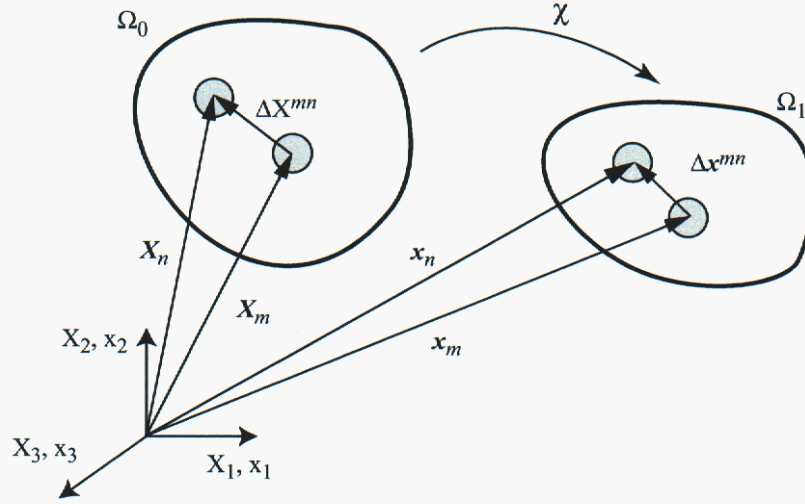


Figure 5.1: General motion in the neighborhood of a discrete atomic particle. The areas labelled Ω_0 and Ω_1 represent the atomic system in reference and current configurations. The shaded circles represent discrete atomic positions that are specified with position vectors \mathbf{X}^m and \mathbf{x}^m in the reference and current configurations.

$t = t_0$ and $t = t_{n+1}$ are denoted by Ω_0 and Ω_{n+1} . The deformation gradient mapping between these configurations is \mathbf{F}_{n+1} . Note that \mathbf{F}_{n+1} is the deformation gradient with respect to the global reference configuration.

Consider the deformation in the neighborhood of atom m that characterized by the changes in the of relative position of its neighbors as shown in figure 5.1. Atom m , is located at the position \mathbf{X}^m in configuration Ω_0 and position \mathbf{x}^m in configuration Ω_{n+1} . In configuration Ω_0 the relative position of neighbor in atom n is given by the vector

$$\Delta \mathbf{X}^{mn} = \mathbf{X}^n - \mathbf{X}^m \quad (5.13)$$

and in configuration Ω_{n+1} by

$$\Delta \mathbf{x}^{mn} = \mathbf{x}^n - \mathbf{x}^m. \quad (5.14)$$

In equations 5.13 and 5.14 the superscripts denote atom numbers, and the vectors $\Delta \mathbf{X}^{mn}$ and $\Delta \mathbf{x}^{mn}$ connect atom m to its neighbors n . There exists a unique linear mapping that transforms the relative position vector $\Delta \mathbf{X}^{mn}$ to $\Delta \mathbf{x}^{mn}$:

$$\Delta \mathbf{x}^{mn} = \mathbf{F} \cdot \Delta \mathbf{X}^{mn}, \quad (5.15)$$

or in index notation

$$\Delta x_i^{mn} = \hat{F}_{ij} \Delta X_j^{mn}. \quad (5.16)$$

Equation 5.15 defines the operation of the discrete deformation gradient at atom m in a manner analogous to the continuum of equation 5.6 in that it maps a vector connecting

atoms m and n in the reference configuration to a vector segment connecting atoms m and n in the current configuration.

We can describe the deformation in the neighborhood of atom m more completely by writing equation 5.6 for each of its neighbors. The set of equations generated by this procedure will not generally be satisfied by a single mapping \mathbf{F} , instead an optimal one is sought.

Consider the the mapping error for an approximate \mathbf{F} :

$$\phi_n = (\Delta \mathbf{x}^{mn} - \mathbf{F} \Delta \mathbf{X}^{mn})^T (\Delta \mathbf{x}^{mn} - \mathbf{F} \Delta \mathbf{X}^{mn}). \quad (5.17)$$

The total, the weighted least squares error among all neighbors of m is given by

$$\phi = \sum_{n=1}^N (\Delta \mathbf{x}^{mn} - \mathbf{F} \Delta \mathbf{X}^{mn})^T (\Delta \mathbf{x}^{mn} - \mathbf{F} \Delta \mathbf{X}^{mn}) w_n \quad (5.18)$$

or in index notation,

$$\phi = \sum_n (\Delta x_k^{mn} - F_{kl} \Delta X_l^{mn}) (\Delta x_k^{mn} - F_{kp} \Delta X_p^{mn}) w_n, \quad (5.19)$$

where N is the total number of neighboring atoms, and w_l is a weight factor that diminishes with the length of $\Delta \mathbf{x}^m$. Minimizing ϕ with respect to the components of \mathbf{F} leads to

$$\frac{\partial \phi}{\partial F_{ij}} = \sum_n (-2 \Delta x_k^{mn} \delta_{ki} \delta_{lj} \Delta X_l^{mn} + \delta_{ki} \delta_{lj} \Delta X_l^{mn} F_{kp} \Delta X_p^{mn} + F_{kl} \Delta X_l^{mn} \delta_{ki} \delta_{pj} \Delta X_p^{mn}) w_n, \quad (5.20)$$

$$\frac{\partial \phi}{\partial F_{ij}} = \sum_n (-2 \Delta x_i^{mn} \Delta X_j^{mn} + 2 \Delta X_j^{mn} F_{ik} \Delta X_k^{mn}) w_n. \quad (5.21)$$

Setting 5.21 equal to zero and solving for the components of \mathbf{F} :

$$\sum_n (-\Delta x_i^{mn} \Delta X_j^{mn} + F_{ik} \Delta X_k^{mn} \Delta X_j^{mn}) w_n = 0 \quad (5.22)$$

$$\sum_n \Delta x_i^{mn} \Delta X_j^{mn} w_n = F_{ik} \sum_n \Delta X_k^{mn} \Delta X_j^{mn} w_n \quad (5.23)$$

or in matrix notation

$$\sum_n \Delta \mathbf{x}^{mn} \Delta \mathbf{X}^{mnT} w_n = \mathbf{F} \sum_n \Delta \mathbf{X}^{mn} \Delta \mathbf{X}^{mnT} w_n \quad (5.24)$$

or

$$\mathbf{F} \mathbf{D} = \mathbf{A} \quad (5.25)$$

in which

$$\mathbf{D} = \sum_l \Delta \mathbf{X}^{ml} \Delta \mathbf{X}^{mlT} w_l, \quad (5.26)$$

and

$$\mathbf{A} = \sum_l \Delta \mathbf{x}^{ml} \Delta \mathbf{X}^{mlT} w_l. \quad (5.27)$$

Then provided that \mathbf{D} was formed from at least three non-coplanar nearest neighbors having nonzero weights the discrete deformation gradient is now determined as

$$\mathbf{F} = \mathbf{A} \mathbf{D}^{-1}. \quad (5.28)$$

Equation 5.28 defines the deformation gradient of Ω_{n+1} with respect to Ω_0 .

5.2.4 Weight Function

The weight function plays an important role in the formulation of the deformation gradient. Generally, we seek for nearest neighbor atoms to have a stronger influence on the computation than those farther away. Thus, the weights should be relatively large for nearest neighbors, and diminish as a function of the distance between the atoms and its neighbors, $w_l = w(\Delta \mathbf{x}^{ml})$. Figure 5.2 shows an example, step weight function. Here the weight function is a set of steps starting with a maximum value of 1 and decreasing to zero. It is important that the step function have a zero spatial gradient to eliminate the influence of thermal vibrations on the computed strains.

With this form, different cutoff distances can be used for various materials to allow for examination of the nonlocal domain of influence.

5.2.5 Discussion

An essential difference between the properties of the discrete deformation gradient defined by equation 5.15 and the continuum form should be noted. The continuum deformation gradient depends on the existence of a body comprised of a continuous distribution of material and a smooth, differentiable motion. These assumptions engender an integrability condition: that the curl of the deformation gradient vanishes. In the discrete system, only a succession of atomic positions is considered, and while the motion of the atom is assumed to be smooth, no postulation of a smooth motion of a continuous region of space is made. Furthermore, the discrete deformation gradient is not known as a function of time and spatial position, but only at atomic positions; hence, no integrability conditions, similar to those of the continuum deformation gradient, are required. On the other hand, the results of this procedure can be viewed as determining the locally linear motion or constant deformation gradient that optimally (in a weighted least-squares sense) approximates the true motion. This is observed from the fact that if the discrete atomic positions are prescribed by a globally linear motion, such as simple shear, then this method will recover the exact deformation gradient. Thus the locally constant deformation gradient satisfies the integrability conditions.

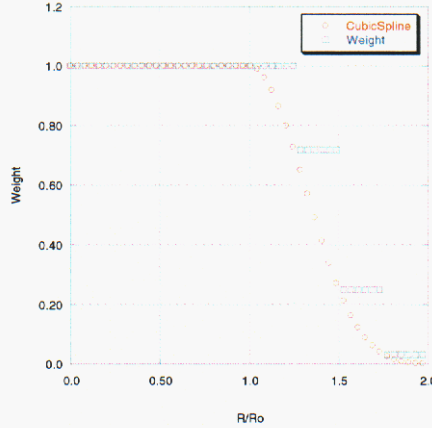


Figure 5.2: An example weight function. The weights used are given by the step function. The values of the weight function are determined by a monotonically decreasing function such as the cubic spline shown. The weight function maintains a value of one for all nearest neighbors then decreases for neighbors farther away.

5.2.6 Example Strain Calculation

This simulation provides an example of the computed strain for a single crystal slab of atoms subjected to fixed-end, simple shear boundary conditions. The slab geometry, shown in Figure 5.3(a), is 35 Å high, 140 Å long, and 4 unit cells thick. The lattice is fcc Nickel with crystal directions $\langle 100 \rangle$, $\langle 011 \rangle$ and $\langle 0\bar{1}0 \rangle$ corresponding to the x , y , and z axes respectively. The boundary conditions applied to the slab include a constant velocity of $v_x = 0.035$ Å/ps and $v_y = 0.0$ on the $+y$, while holding the $-y$ surface atoms fixed as shown in Figure 5.3(b). Periodic boundary conditions are applied in the z direction. The velocities of the interior atoms were initialized using a Boltzmann distribution at 300K, and the system was equilibrated to accommodate any surface relaxation on the $\pm x$ surfaces before the shear velocity was applied. Note that these boundary conditions differ from a rigorous continuum description of simple shear in that the x faces are free to deform and do not remain parallel during the simulation and the prescribed shear velocity is only applied to the atoms on the

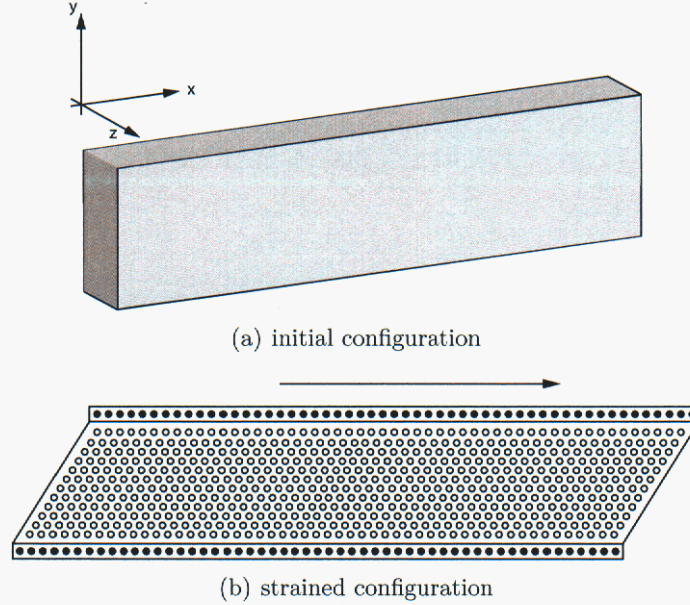


Figure 5.3: Schematic of simulation block of atoms at the (a) initial configuration and (b) at large strain in which the clear circles represent active atoms and the dark circles represent boundary atoms.

$+y$ surface. Because the interior atoms were not initialized with a superimposed $x - dir$ velocity, a shock is introduced into the block of material when the shear velocity is applied.

The system stress, Green strain and Almansi strain tensors were computed for this simulation. The stress was computed using the viral stress definition given by equation 6.1. The strain tensors were computed according to equations 5.11 and 5.12 using the deformation gradient of 5.28.

The results of the simulation are summarized in Figures 5.4 to 5.6. The shear strain averages are shown in Figure 5.4, and the percent difference in the average Green and Almansi shear strains are shown in Figure 5.5.

Qualitatively, Figure 5.4 shows that, the average Green and Almansi shear strains increase over the course of the simulation at a rate that corresponds well with a simple shear response. At 8% strain the system attains a maximum stress and from this point the average Green and Almansi strain values begin to diverge, with the Almansi strain increasing at a lower rate than the Green.

Figure 5.5 shows Green and Almansi shear stress values have an initial difference of approximately 3% that quickly decreases to approximately 1%. This is likely due to the initial shock response of the material. At 4% applied strain, a jump in the in the shear strain values occurs. This point corresponds to a slight drop in slope of the stress-strain curve which indicative of dislocation nucleation. At this same time the difference between strains jumps to approximately 10%. The difference then decreases slowly until another jump occurs

at the system's yield point of 8% strain. After 10% strain, the difference between the Green and Almansi strain measures increases apart from changes in the stress state of the system.

The local atomic strains are shown in Figures 5.7 and 5.6. These figures show that the strain is oriented with the crystallographic planes. The Green strain show most activity along the $\langle 111 \rangle$ planes, while the Almansi shows activity along the $\langle \bar{1}11 \rangle$.

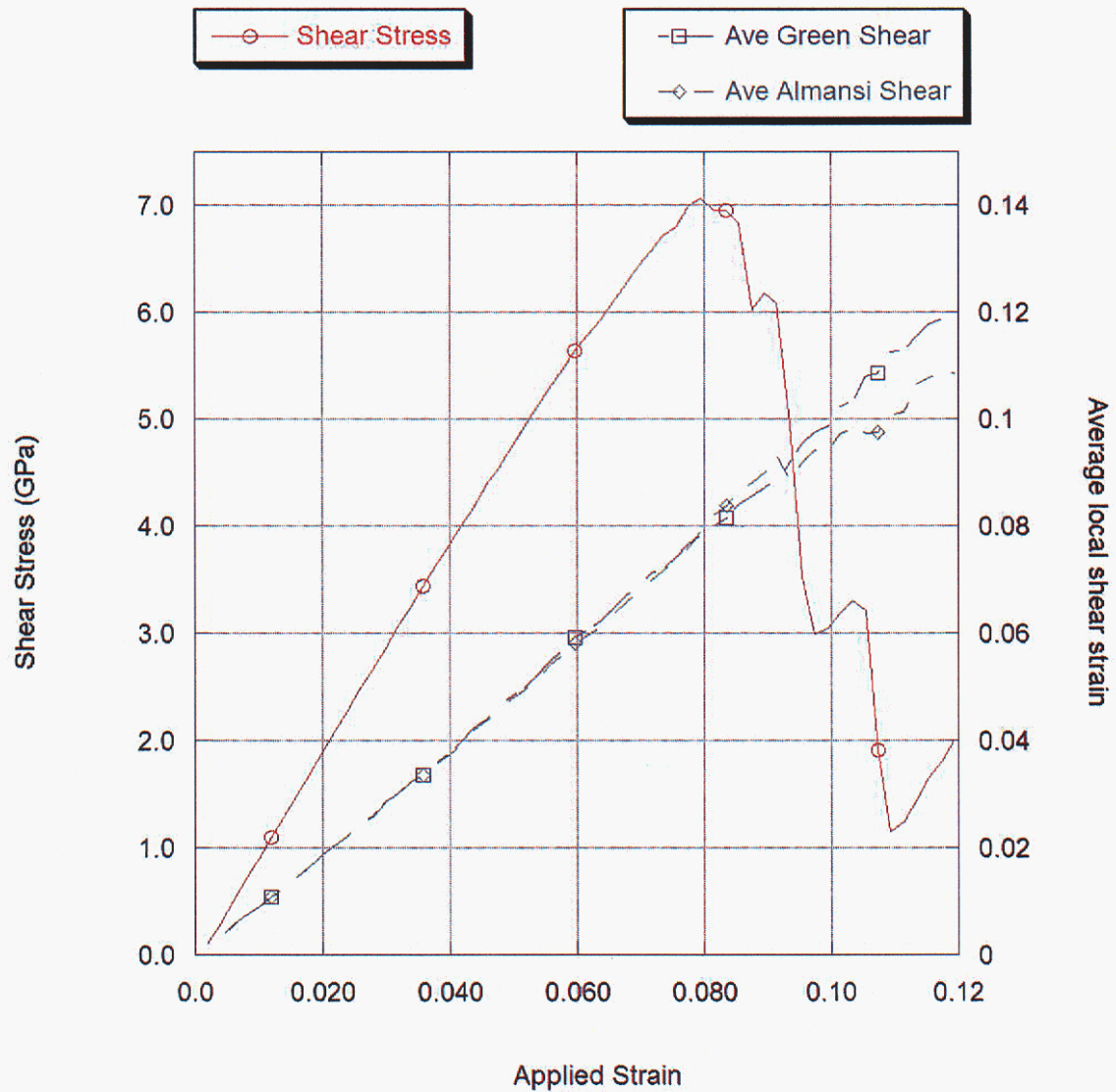


Figure 5.4: The computed average local Green and Almansi shear strains along with the average shear stress are plotted vs applied strain.

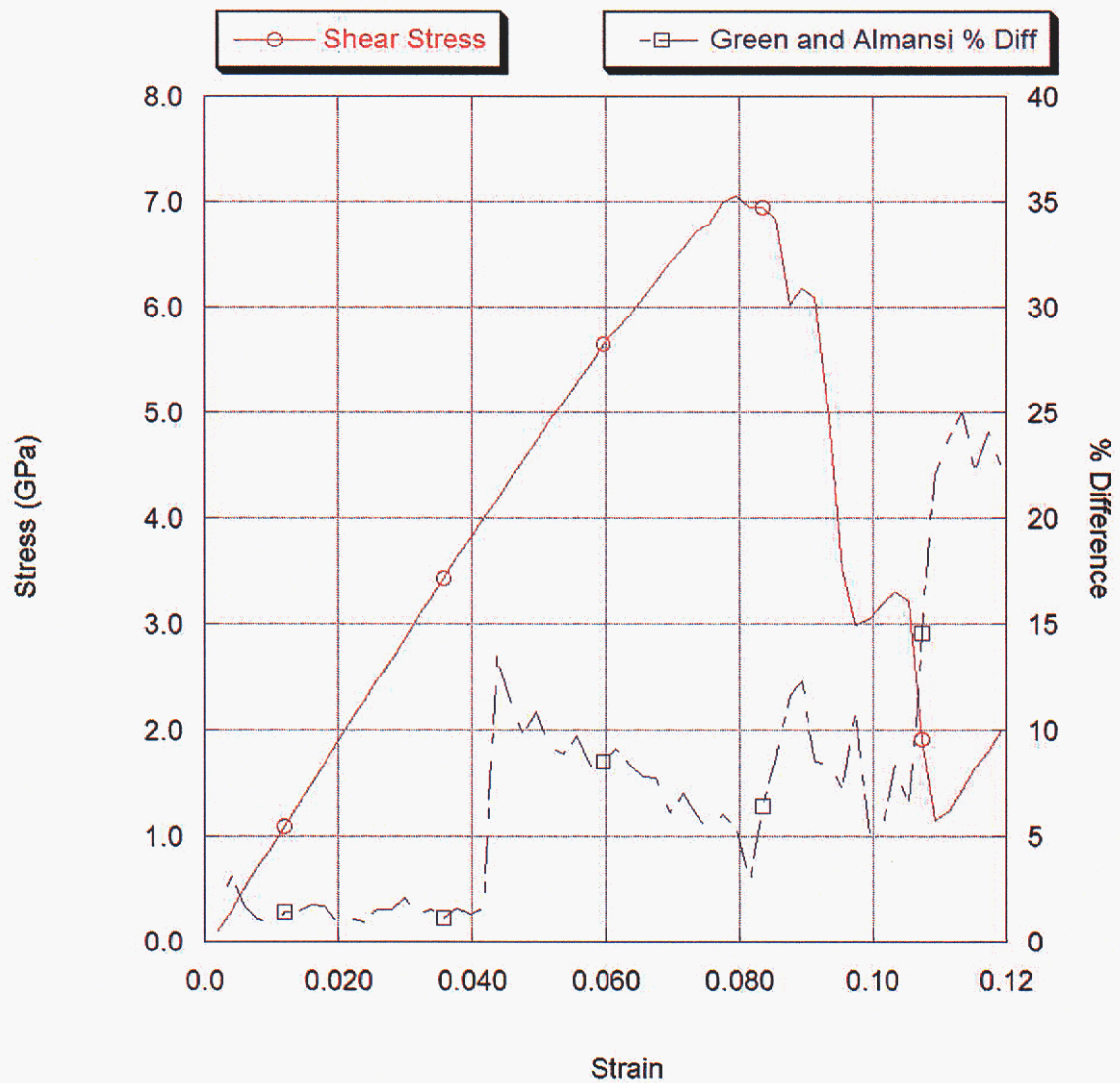


Figure 5.5: The percent difference in the computed average local Green and Almansi shear strains are plotted vs applied strain.

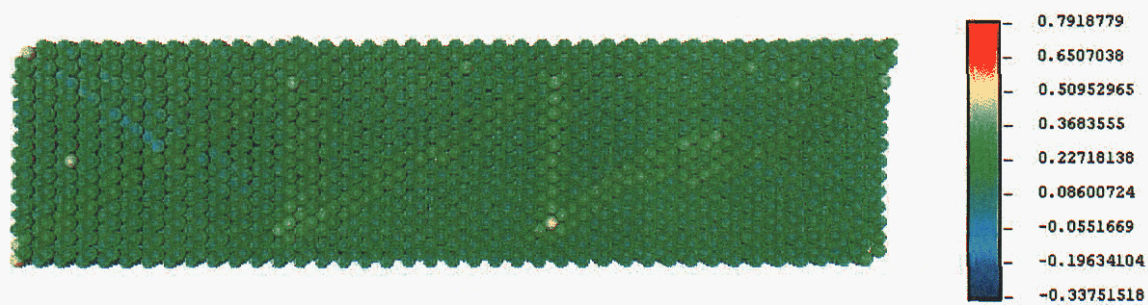


Figure 5.6: Almansi strain at 12% strain.

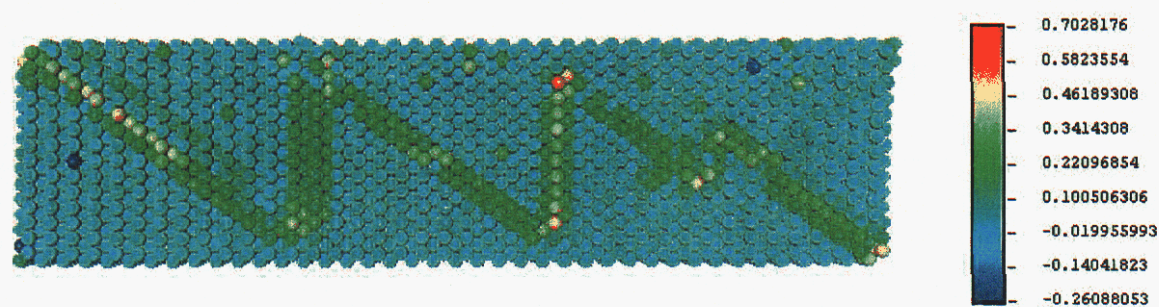


Figure 5.7: Green strain at 12% strain.

Chapter 6

Computational Example

6.1 Introduction

It was found in work by Horstemeyer et al. [1] that the failure of a material simulated with MD is dependent on the length scale of the material. Specifically, the yield strength of a material decreases with increasing size. This trend was seen in a range of specimen sizes from about 2 nm to $2\text{ }\mu\text{m}$. All of these specimens were single crystals without initial defect, and failure initiated by the emission of dislocations at the corners of the rectangular specimens.

In real materials, ductile failure occurs due to the nucleation, growth and coalescence of voids. In order to use MD to simulate this process, it is important to first understand what the effects of specimen size might be on these void dynamics. As a first step toward this understanding, we have explored the effects of length scale on void growth under uniaxial tensile strain.

6.2 Problem Statement

The geometry used in our simulations is shown in Figure 6.1. A thin rectangular slab of nickel with a void at its center is strained uniaxially in the x direction. The test section has length $2L_x$ and $2L_y$ in the x and y directions, and thickness L_z . The void radius is R_v .

6.2.1 Atomic lattice

We use an atomic representation of nickel using a MEAM potential. The lattice spacing of nickel is $a = 3.52\text{ }\text{\AA}$, giving an equilibrium atomic spacing of $r_e = 2.49\text{ }\text{\AA}$. The crystal directions $\langle 100 \rangle$, $\langle 011 \rangle$ and $\langle 0\bar{1}1 \rangle$ correspond to the x , y , and z axes, respectively.

6.2.2 Boundary conditions

The uniaxial strain is applied by imposing a constant velocity $v_x = \pm \dot{\epsilon} L_x$ on one unit cell-thickness of atoms on the $\pm x$ surfaces. Periodic boundary conditions are applied in the y

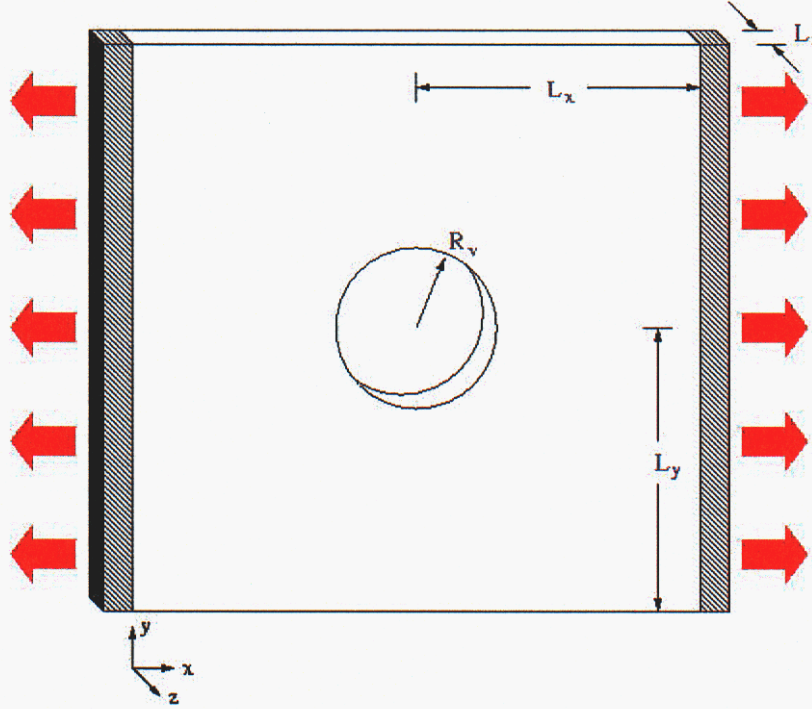


Figure 6.1: Geometry used for uniaxial tension void growth tests.

and z directions. Note that this implies zero strain in the y and z directions, so that the condition is one of uniaxial strain (and not necessarily uniaxial stress).

6.2.3 Initial conditions

Before the uniaxial strain is applied, the system is run for $10ps$ at a constant temperature of $300K$ for equilibration. A Nosé-Hoover thermostat [17] is used to enforce constant temperature. When the equilibration phase is finished, a linear x velocity profile $v_x = \epsilon x$ is applied to all atoms. This is done in order to avoid an initial shock wave, which otherwise would result from the instantaneous application of the boundary conditions on the $\pm x$ surfaces.

6.2.4 Run cases

Four cases are run in this study, summarized in Table 6.1. The cases have increasing values of L_x , L_y , and R_v such that the ratios between these quantities (and therefore the initial void volume fractions) are constant throughout the cases. The thickness L_z is kept constant throughout the cases. The number of atoms in the simulations ranges from about 11,000 in the smallest run to about 642,000 in the largest.

Run	L_x (nm)	L_y (nm)	L_z (nm)	R_v (nm)	atoms	$\dot{\epsilon}$ (s ⁻¹)
1	5.3	5.2	1.0	0.7	11K	2×10^8
2	10.6	10.5	1.0	1.4	41K	2×10^8
3	21.1	20.9	1.0	2.8	162K	2×10^8
4	42.2	41.8	1.0	5.6	642K	2×10^8

Table 6.1: Summary of cases run

6.2.5 Data collection

A number of quantities are computed and output periodically throughout the simulations. These include stress, void fraction, and centro-symmetry parameter.

The stress is computed according to the virial stress definition:

$$\sigma_{ij} = \left\langle \frac{1}{V_\alpha} \sum_\beta F_{\alpha\beta}^i r_{\alpha\beta}^j \right\rangle_\alpha \quad (6.1)$$

where i and j are indices representing spatial directions, α and β are atom indices, and the notation $\langle \cdot \rangle_\alpha$ represents an average over atoms α . In the region near the void, the reduced $y - z$ cross-section of the specimen is expected to lead to an increased stress. This increased stress, however, does not correspond to an increase in the macroscale stress that would be measured in the continuum; it is a result of the microscale geometry only. In order to avoid this unwanted stress increase, the average in equation 6.1 is taken only over atoms with an initial x position such that $|x| \geq 2R_v$, so that atoms close to the void (in the x direction) are not counted.

The void fraction is computed by discretizing the computational domain into a grid of cells of size slightly larger than the equilibrium spacing of the atoms r_e . The volume of the solid is then approximated by counting the number of cells that contain atoms, while the volume of the void is computed as the number of cells interior to the material that have zero atoms. The “interior of the material” is defined by examining each row of cells in the x direction; the interior is the range of cells lying between the extreme atom x positions in that row. Thus any empty cell that has non-empty cells beyond it in both the $+x$ and $-x$ directions is considered part of the void. With this definition, we can account for void volume created through the nucleation of new voids, but do not count the empty space beyond the boundaries of the specimen in the x direction.

The centro-symmetry parameter of a given atom provides a measure of the level of disturbance of that atom’s environment from the symmetric crystal structure. The formula for the parameter for an FCC crystal is [18]:

$$C_\alpha^{FCC} = \sum_{\beta=1}^6 |\mathbf{r}_{\alpha,\beta} - \mathbf{r}_{\alpha,\beta+6}|^2 \quad (6.2)$$

The summation in this expression is taken over the six pairs of opposing neighbors of an atom. For an atom in a perfect FCC crystal at equilibrium, the centro-symmetry parameter is zero. A larger value is computed near surfaces, grain boundaries and defects. By plotting the atoms with a centro-symmetry parameter larger than some cutoff value (usually 2.0), we can visualize the dislocation structure of the deforming material.

6.3 Results

A series of snapshots for each run is shown in Figures 6.2 through 6.5. Qualitatively, we see that several similarities exist across all runs. Initially, all deformation is elastic, and no dislocations can be seen. The beginnings of plastic failure are seen when dislocations are emitted from the void surface. The first dislocations propagate along the close-packed planes of the material, in this case the $\langle 111 \rangle$ planes at an angle of $\pm 35.3^\circ$ to the x axis. The dislocations form an “X” pattern on these planes before spreading to fill most of the material. The void grows in a symmetric elliptical shape early in the deformation before becoming much more irregular; most of the void growth occurs in this irregular shape.

Several differences between the runs are also apparent. For the larger runs, dislocations are emitted at a lower strain than for smaller runs, indicating that plastic deformation occurs sooner in the larger runs. For example, in run 1 no propagation of dislocations is seen until between 8% and 12% strain, while for run 4 it is clear that dislocations have already propagated at 4% strain. Another very interesting phenomenon is seen only in the largest run, case 4. At around 9% strain, two small new voids nucleate slightly away from the surface of the central void. As the strain increases, these voids grow and eventually coalesce with the larger void. This behavior is not seen in any of the other simulations.

More quantitative comparisons can be made by looking at the stress-strain curves in Figure 6.6. Immediately after the equilibration stage ($t = 0$ in the figure), there is a small compressive stress due to the temperature expansion (the initial configuration is at equilibrium at $T = 0K$), and this stress is equal for all four simulations. When the tensile strain is applied, initially all deformation is elastic with an elastic modulus that is constant across the simulations.

Differences among the simulations begin to occur at around 4% strain. At this point, the largest simulation deviates from the elastic behavior; this corresponds to the dislocations seen being emitted from the void surface in Figure 6.5(b). This yield point happens at a larger strain (and stress) for each of the smaller simulations in turn. Thus we see that yield strength increases as specimen size decreases, which is identical to the results of Horstemeyer et al [1].

After yielding, the smallest simulation shows a sharp drop in stress such as might be seen in brittle failure. However, the stress does not drop completely to zero at this point, but we see an overall downward trend characterized by periods of slow increase interspersed with sharp decreases. Thus, this nanoscale failure appears to be discrete in nature. As the simulations get larger, the drop in stress at yield becomes less precipitous, and fluctuations become smaller. For the largest simulation, run 4, the behavior is similar to that of

macroscale plastic failure, with a smooth transition from elastic to plastic deformation; this is apparently the result of the discrete dislocation motions seen in run 1 being “averaged out” over these larger scales.

The void growth curves for the four simulations are plotted in Figure 6.7. The onset of fast void growth for each simulation in this figure corresponds to the yield point in Figure 6.6. In general, it is expected that the plastic deformation will be incompressible, and therefore volume conserving. Therefore, the total change in volume of the material plus void will be due to void growth alone. By this geometric argument, the void volume fraction is expected to be approximately a function of the strain alone:

$$V_f = \frac{V_{f,0} + \epsilon_p}{1 + \epsilon_p} \quad (6.3)$$

where V_f is the void fraction, $V_{f,0}$ is the initial void fraction, and ϵ_p is the plastic strain. In Figure 6.8, this expression is plotted along with the simulated void fractions, where now the x axis shows an approximation of the plastic strain, defined simply as the total strain minus the yield strain estimated from Figure 6.6. The four curves overlay this line fairly well.

6.4 Discussion

As in previous studies, these simulations of void growth confirm that yield strength is a function of specimen size at these very small scales. As size increases, yield strength increases, and stress-strain behavior begins to approach that of the continuum as fluctuations are averaged out over many atoms.

However, it appears that other than the change in yield strain, there is not a large effect of length scale on void growth. In Figure 6.8, showing void growth vs. plastic strain, all four curves approximately overlay the theoretical line that is derived from geometrical arguments alone. Thus we conclude that for these single-void simulations, once plastic deformation initiates, void growth behavior is universal across the length scales studied. Nothing can be said at this point, however, about length scale effects on void nucleation or coalescence.

One interesting phenomenon seen here is the nucleation of smaller voids away from the surface of the large void in Figure 6.5. This is an unexpected result that is not normally accounted for in void growth models. Further studies should be undertaken to determine whether this is a real phenomenon, or an artifact of some artificial aspect of our simulations such as the very high strain rate.

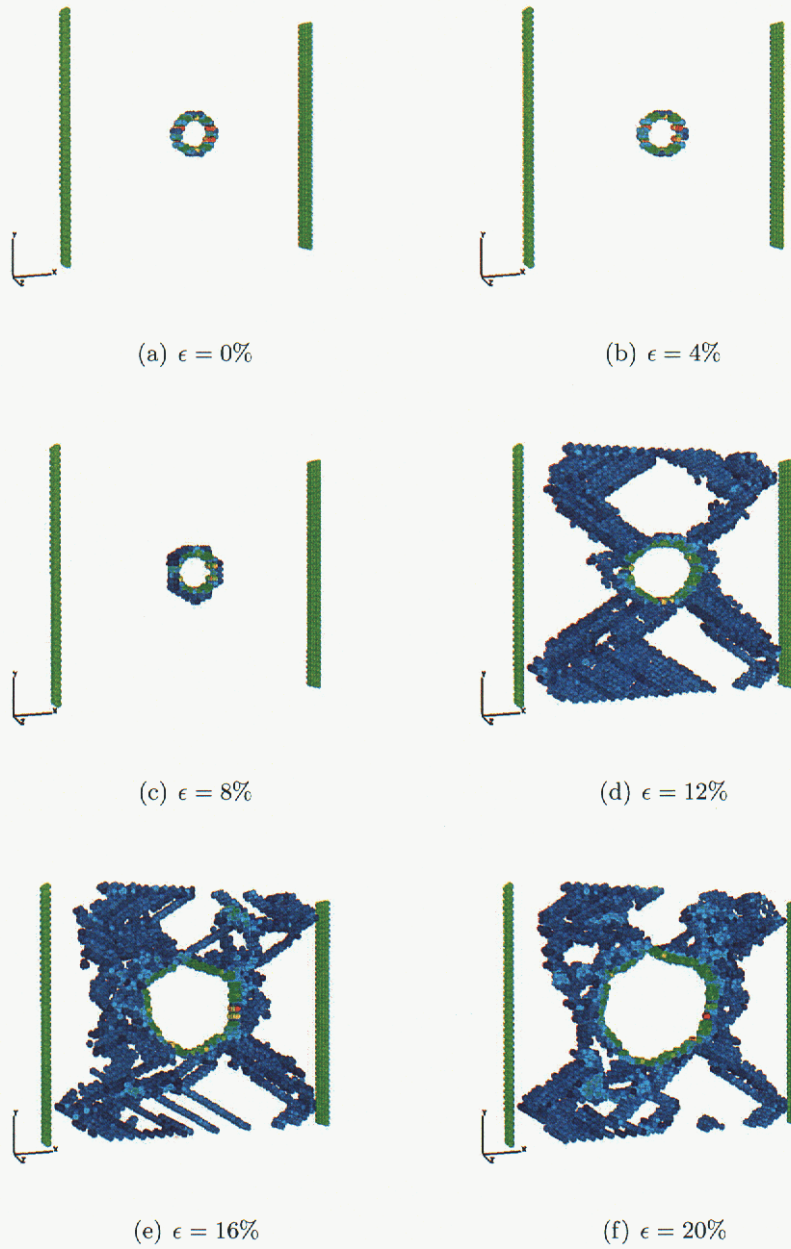


Figure 6.2: Development of dislocation structure for run 1.

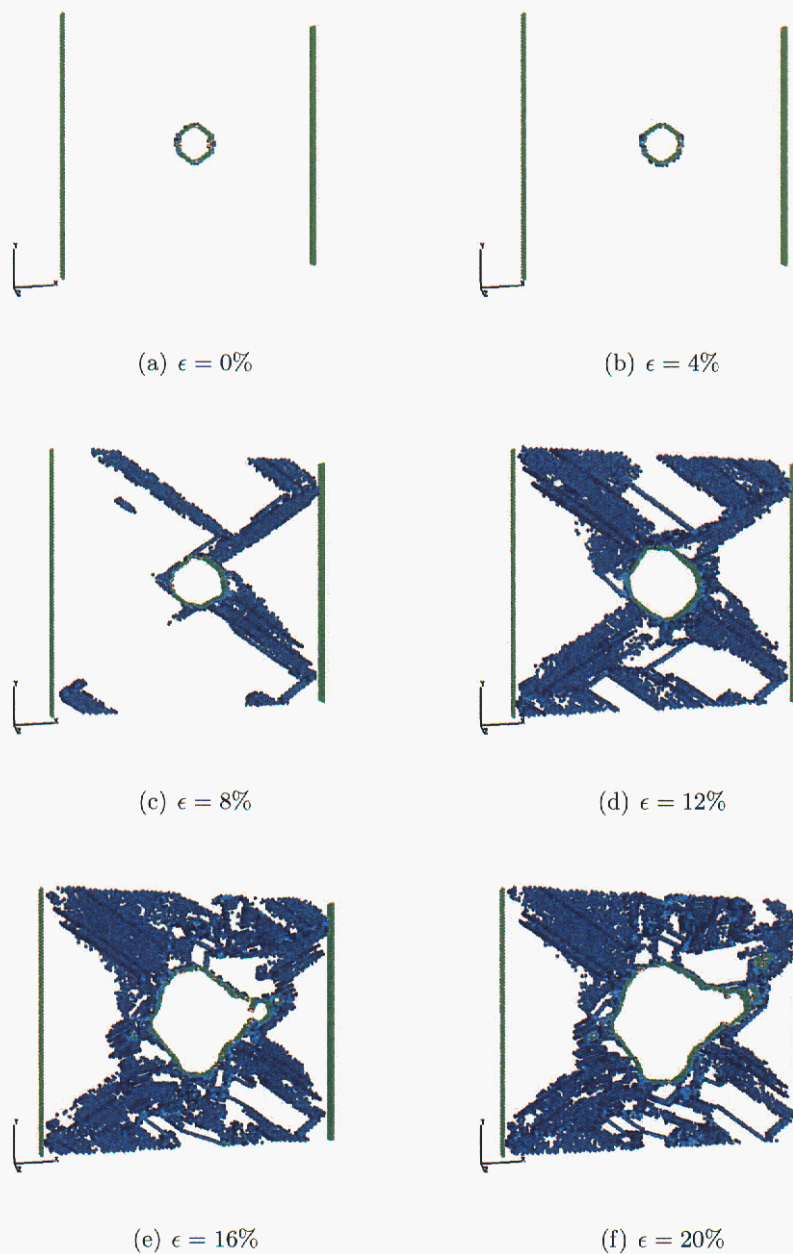


Figure 6.3: Development of dislocation structure for run 2.

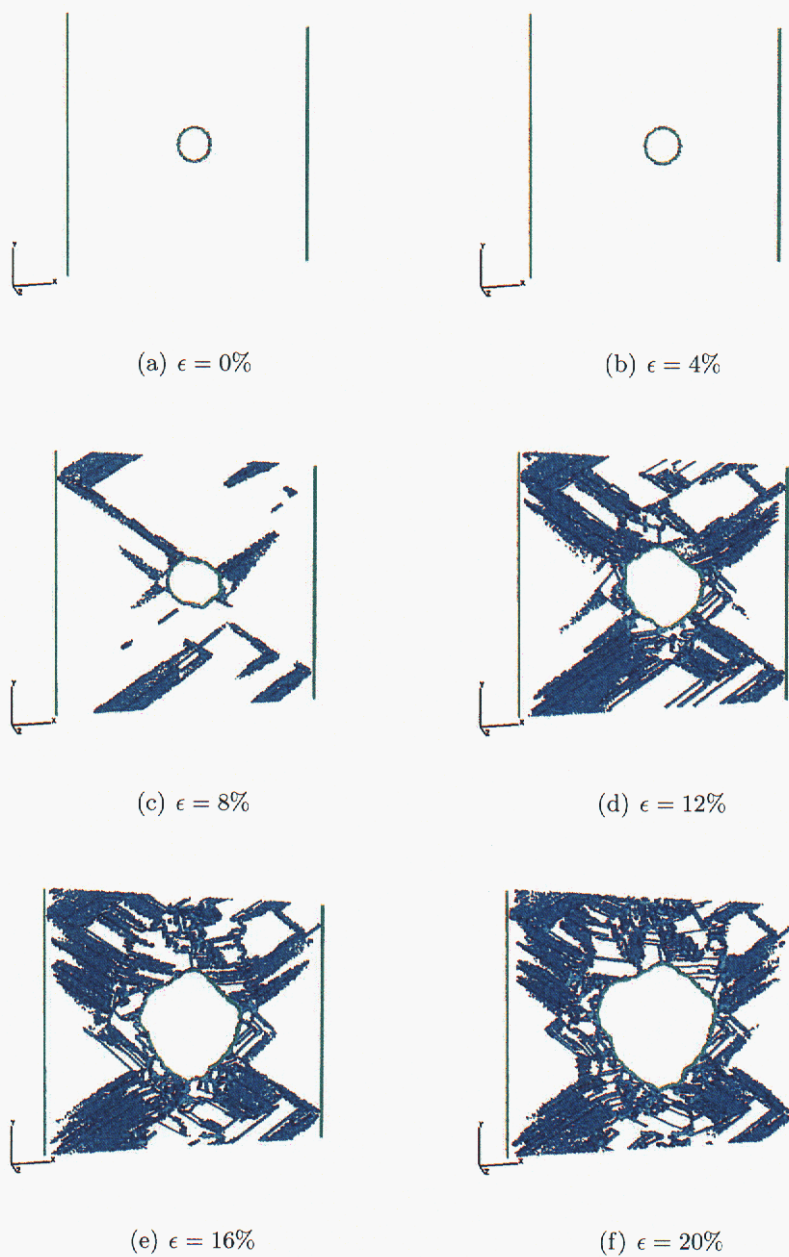


Figure 6.4: Development of dislocation structure for run 3.

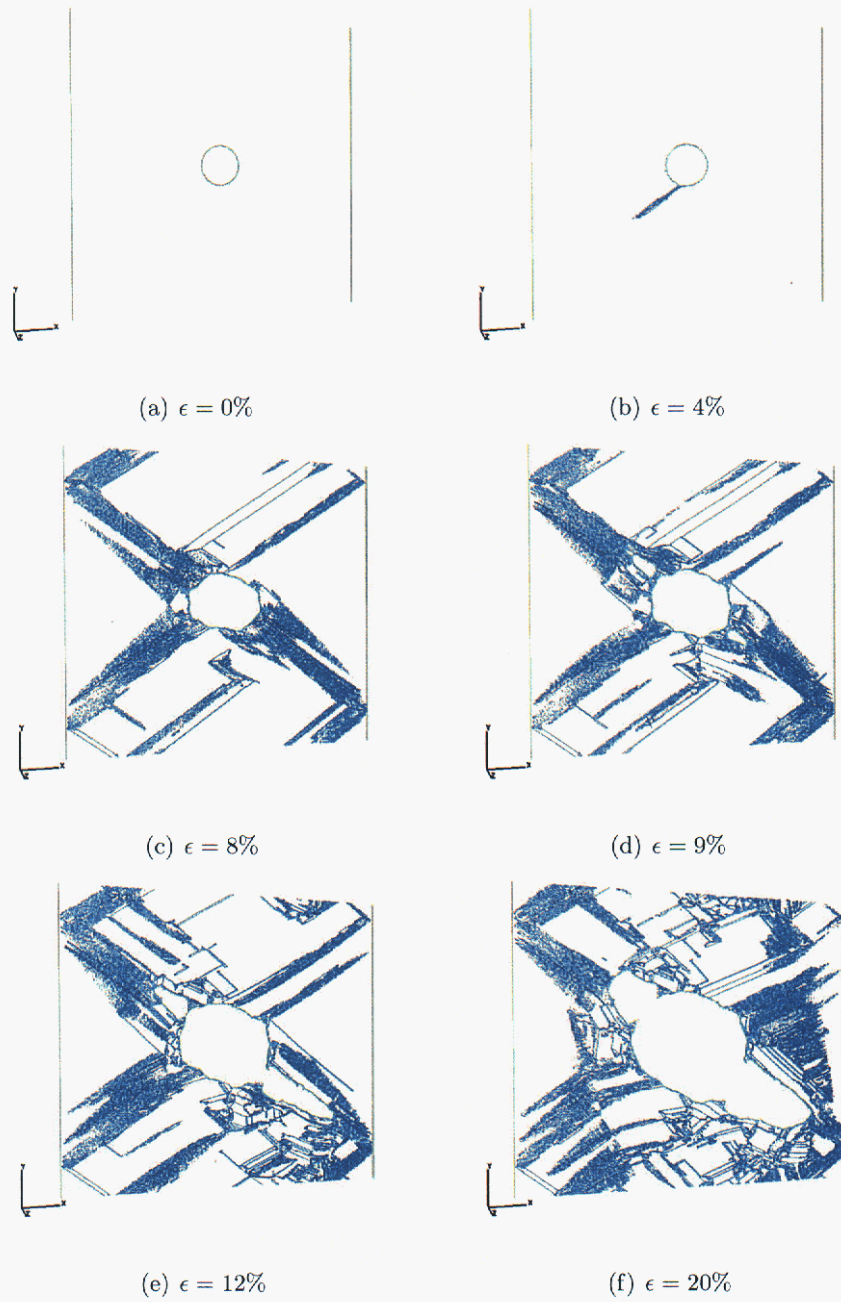


Figure 6.5: Development of dislocation structure for run 4.

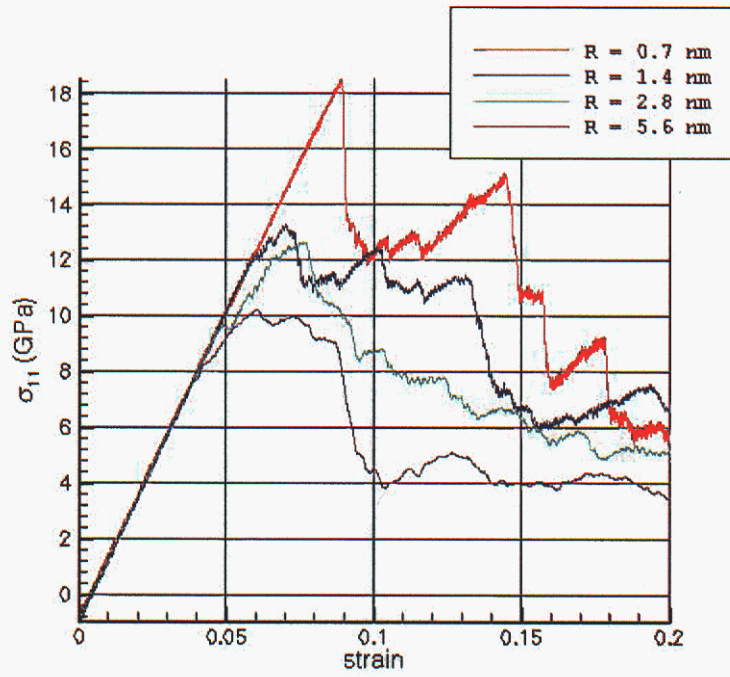


Figure 6.6: Stress-strain behavior of the four simulations.

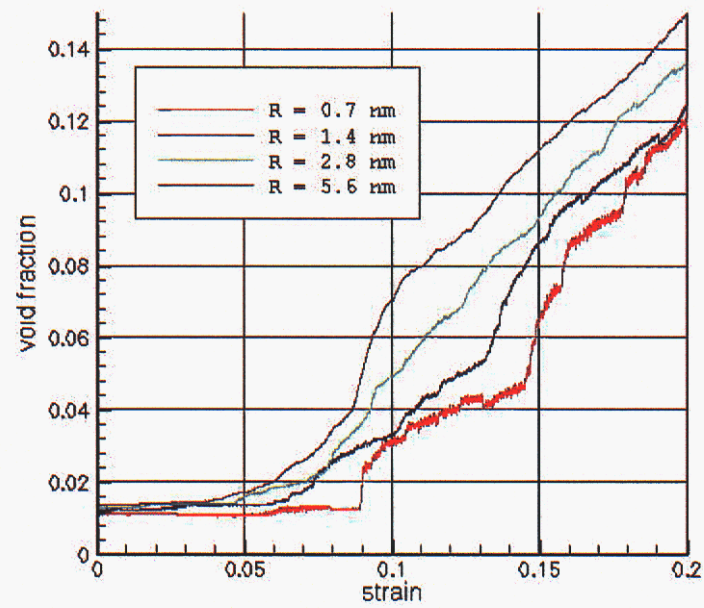


Figure 6.7: Void fraction vs. strain.

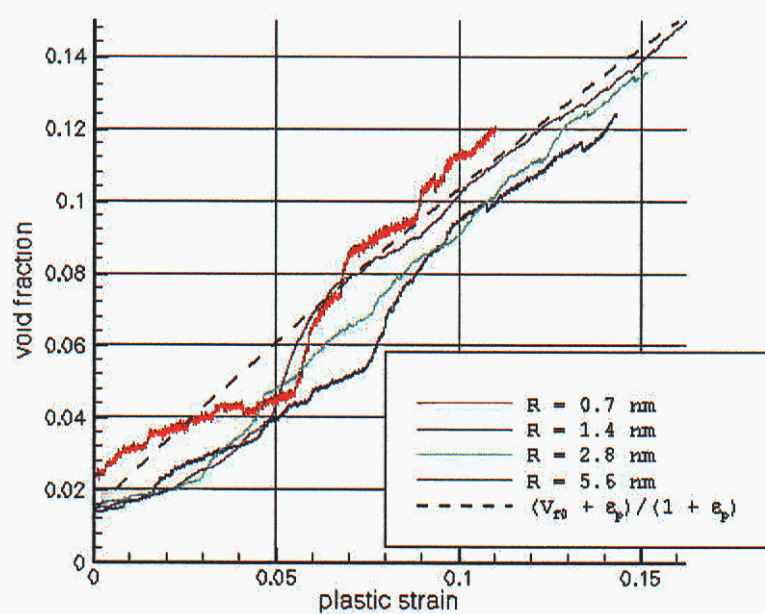


Figure 6.8: Void fraction vs. plastic strain.

This page intentionally left blank.

Bibliography

- [1] M. F. Horstemeyer, M. I. Baskes, and S.J. Plimpton. Length scale and time scale effects on the plastic flow of fcc metals. *Acta Materialia*, 49(20):4363–4374, 2001.
- [2] M. F. Horstemeyer, M. I. Baskes, and S. J. Plimpton. Computational nanoscale plasticity simulations using embedded atom potentials. *Theoretical and Applied Fracture Mechanics*, 37(1-3):49–98, 2001.
- [3] Da Chen. Structural modeling of nanocrystalline materials. *Computational Materials Science*, 3:327–333, 1995.
- [4] Ralph Wyckoff. *Crystal structures*. New York, Interscience Publishers, 2nd edition, 1963-1971.
- [5] US Naval Research Laboratory. Crystal lattice structures. “<http://cst-www.nrl.navy.mil/lattice/index.html>”, 2003.
- [6] A.J.C. Ladd and L.V. Woodcock. Interfacial and coexistence properties of lennard-jones system at triple point. *Molecular Physics*, 36(2):611–619, 1978.
- [7] Murray S. Daw and M. I. Baskes. Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals. *Physical Review B*, 29(12):6443–6453, 1984.
- [8] S Plimpton. Fast parallel algorithms for short-range molecular-dynamics. *Journal Of Computational Physics*, 117(1):1–19, 1995.
- [9] S. Plimpton and B. Hendrickson. A new parallel method for molecular dynamics simulation of macromolecular systems. *Journal Of Computational Chemistry*, 17(3):326–337, 1996.
- [10] G.S. Heffelfinger and M.E. Lewitt. A comparison between two massively parallel algorithms for monte carlo computer simulation: An investigation in the grand canonical ensemble. *Journal Of Computational Chemistry*, 17(2):250–265, 1996.
- [11] M. I. Baskes. Modified embedded-atom potentials for cubic materials and impurities. *Physical Review B*, 46(5):2727–2742, 1992.

BIBLIOGRAPHY

- [12] H. C. Huang, N. M. Ghoniem, J. K. Wong, and M. I. Baskes. Molecular-dynamics determination of defect energetics in beta-SiC using 3 representative empirical potentials. *Modelling Simul. Mater. Sci. Eng.*, 3(5):615–627, 1995.
- [13] M. I. Baskes. Determination of modified embedded atom method parameters for nickel. *Mater. Chem. Phys.*, 50(2):152–158, 1997.
- [14] Peter H. Mott, Ali S. Argon, and Ulrich W. Suter. The atomic strain tensor. *Journal of Computational Physics*, 101:140–150, 1992.
- [15] M. L. Falk. Molecular-dynamics study of ductile and brittle fracture in model noncrystalline solids. *Physical Review B*, 60(10):7062–7070, 1999.
- [16] R W Ogden. *Non-linear elastic deformations*. Dover, 1984.
- [17] D. Frenkel and B. Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, San Diego, 2nd edition, 2002.
- [18] C. L. Kelchner, S. J. Plimpton, and J. C. Hamilton. Dislocation nucleation and defect structure during surface indentation. *Phys. Rev. B*, 58(17):11085–11088, 1998.

Chapter 8

Distribution

- 5 Mark Horstemeyer
Mechanical Engineering
Mississippi State University
206 Carpenter Bldg.
PO Box ME
Mississippi State, MS 39762
- 1 Hongbing Fang
Center for Advanced Vehicular Systems
Box 9627
Mississippi State University
Mississippi State, MS 39762-9627
- 1 Mike I. Baskes
MST-8, MS G755
Los Alamos National Laboratory
Los Alamos, NM 97545
- 1 Mo Li
Georgia Institute of Technology
Materials Science and Engineering
771 Ferst Drive
N.W. Atlanta, GA 30332-0245
- 10 MS 0316 A. Slepoy, 9235
- 1 J. B. Aidun, 9235
- 1 MS 1111 S.J. Plimpton, 9212

CHAPTER 8. DISTRIBUTION

1 MS 1411 J.J. Hoyt, 1834

1 MS 9001 M.E. John, 8000
Attn:
R.H. Stulen, 8100, MS 9004
D.R. Henson, 8200, MS 9007
W.J. McLean, 8300, MS 9054
P.N. Smith, 8500, MS 9002
K.E. Washington, 8900, MS 9003

10 MS 9042 P.M. Gullett, 8774
1 P.A. Spence, 8774

10 MS 9042 G.J. Wagner, 8752
1 C. Moen, 8752

1 MS 9161 E.P. Chen, 8763
1 P.A. Klein, 8763
1 C.J. Kimmer, 8763
1 J.A. Zimmerman, 8763

1 MS 9405 K.L. Wilson, 8770
Attn:
T.J. Shepodd, 8762, MS 9403
C.H. Cadden, 8772, MS 9402
J.C.F. Wang, 8773, MS 9403

1 MS 9403 J.M. Hruby, 8700
Attn:
G.D. Kubiak, 8750, MS 9404
W.R. Even, 8760, MS 9161

1 MS 9405 S. Aubry, 8763
1 D.J. Bammann, 8763
1 R.E. Jones, 8763

1 MS 0323 D.L. Chavez, LDRD Office, 1011
1 MS 9018 Central Technical Files, 8945-1
2 MS 0899 Technical Library, 9616
1 MS 0612 Review & Approval Desk, 9612
For DOE/OSTI via URL